

国际信息工程先进技术译丛

WILEY

IP地址管理原理与实践

IP Address Management Principles and Practice

(美) Timothy Rooney 编著
陈海英 袁开银 王玲芳 等译





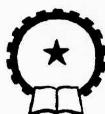
机械工业出版社
CHINA MACHINE PRESS

国际信息工程先进技术译丛

IP 地址管理原理与实践

（美）Timothy Rooney　编著
陈海英　袁开银　王玲芳　等译





机械工业出版社



图书在版编目(CIP)数据

IP 地址管理原理与实践/(美)鲁尼(Rooney, T.)编著;陈海英,袁开银,王玲芳等译. —北京:机械工业出版社,2013.1
(国际信息工程先进技术译丛)
书名原文:IP Address Management Principles and Practice
ISBN 978-7-111-40870-3

Ⅰ.①I… Ⅱ.①鲁…②陈…③袁…④王… Ⅲ.①互联网络-网址
Ⅳ.①TP393.409.2

中国版本图书馆 CIP 数据核字(2012)第 301103 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑:张俊红 责任编辑:林 桢 版式设计:赵颖喆
责任校对:常天培 封面设计:马精明 责任印制:乔 宇
北京机工印刷厂印刷(三河市南杨庄国丰装订厂装订)
2013 年 3 月第 1 版第 1 次印刷
169mm × 239mm · 22.5 印张 · 503 千字
0 001—3 000 册
标准书号:ISBN 978-7-111-40870-3
定价:89.80 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务
社服务中心:(010)88361066
销售一部:(010)68326294
销售二部:(010)88379649
读者购书热线:(010)88379203

网络服务
教材网:http://www.cmpedu.com
机工官网:http://www.cmpbook.com
机工官博:http://weibo.com/cmp1952
封面无防伪标均为盗版

本书介绍了将网络管理科学应用到互联网协议地址空间及相关联的网络服务等内容。本书分为四个部分，第Ⅰ部分给出IPv4、IPv6以及IP地址分配和子网划分技术综述；第Ⅱ部分描述用于IPv4和IPv6的DHCP，并解释依赖于DHCP的各项应用、DHCP服务器部署策略以及DHCP和相关的网络接入安全；第Ⅲ部分描述DNS协议、DNS应用、部署策略和相关联的配置以及安全性；第Ⅳ部分描述以聚合方式管理IP地址空间的各项技术和日常IP地址管理功能。

本书可作为IP网络规划人员、工程师和管理人员的实践用书，同时可供部署IPv6网络的技术人员参考。

本书版权登记号：图字01-2011-7142。

译者序

TCP/IP协议族是因特网的核心组件，对于这个发展到成熟阶段的全球网络而言，能够发展到今天，其起到了功不可没且至关重要的作用。有人说，从体系架构方面而言，因特网已经开始出现制约其发展的瓶颈，这主要是针对IP这个细腰来说的，认为它太细了，不能承载太多的来自网络层以上的多样化需求。但无论如何说，我们在目前还看不到任何技术可在短期内能够替代IP成为下一代全球网络的支撑性关键技术。因特网的一个核心理念是端到端，这从IP报文格式中的源地址和目的地得以体现。IP地址是因特网的核心战略资源，无论是IPv4，还是IPv6，即使可用的地址池再大，面临日益复杂的巨型网络，其前景都是令人不容乐观的，所以IP地址的管理是维护网络正常运行、发展和演化的基础。

本书分为四个部分。本书前三部分的焦点分别是三项核心IPAM功能：IP寻址和管理、DHCP以及DNS。第Ⅳ部分集成这三部分，描述管理技术和实践。

第Ⅰ部分给出IPv4、IPv6以及IP地址分配和子网划分技术的详细综述；第Ⅱ部分给出IPv4 DHCP和IPv6 DHCP的概述，并讲解依赖于DHCP的各项应用、DHCP服务器部署策略以及DHCP和相关的网络访问安全；第Ⅲ部分描述DNS协议、DNS应用、部署策略和相关联的配置以及安全性（包括DNS服务器和配置的安全以及DNS-SEC）；第Ⅳ部分整合前三部分，讨论逻辑严密、协调一致地管理IP地址空间的各项技术，其中论及对DHCP和DNS的影响。

本书由王玲芳负责第1和2章翻译以及全书译稿的统稿、校对等，袁开银负责第3~10章的翻译工作，陈海英负责第11~15章、词汇表、RFC索引的翻译工作。本书在翻译过程中，吴秋义、李冬梅、潘东升、吴璟、王弟英、李虹、游庆珍、李传经、吴昊、李睿、刘学录、马安华、陈忠原、赵妍、费岚、李志刚、李岩、张瑞等同志参加了部分的翻译工作，在此表示感谢。另外，感谢互联网领域的先驱者、实践者和研究人员。

不过，需要指出的是，本书的内容仅代表作者个人的观点和见解，并不代表译者及其所在单位的观点。另外，由于翻译时间比较仓促，疏漏错误之处在所难免，敬请读者原谅和指正。

译　者

原书前言

IP地址管理（IPAM）实践内容包括将网络管理学科知识应用到因特网协议（IP）地址空间和相关联的网络服务（即动态主机配置协议（DHCP）和域名系统（DNS））。IP地址规划和DHCP服务器、DNS服务器配置之间的联系是不可分的。一个IP地址的改变会影响DNS信息，也许还会影响DHCP。这些服务提供了如今融合服务IP网络的基础，该网络可提供独特的在任何时间、任何地点的通信能力。

如果如笔记本计算机（膝上机）或IP语音（VoIP）电话等端用户设备不能通过DHCP得到一个IP地址，那么这些设备将是不可用的，此时用户将会给服务台打电话。类似地，如果没有正确配置DNS，则依据名字、电话号码或网页地址导航的应用将同样地受到影响，并导致服务台接到大量呼叫电话。

在一个企业或服务提供商的IP网络管理策略中，有效的IPAM实践是一项核心构成。如此，IPAM地址配置、变更控制、审计、报告、监视、故障解决和有关功能，都适用于以下三项基础IPAM技术。

（1）IP地址子网划分和记录跟踪（IPv4/IPv6寻址）：一项周密的IP地址规划的维护，这可提升路由汇总、维护准确的IP地址清单，并提供一项自动的各IP地址指派和记录跟踪机制。在每个子网上各IP地址指派的这项记录跟踪措施，包括由硬编码指派的那些地址（例如路由器或服务器）以及其他动态指派的那些地址（例如笔记本计算机和VoIP电话）。

（2）DHCP：与位置和设备类型有关的自动IP地址和参数指派。这要求对配置于设备上的地址指派记录跟踪，以及预留动态分配的地址池。为了支持设备请求一个IP地址，并相应地接收到一个位置相关的地址，可将这些地址配置于DHCP服务器上。

（3）DNS：主机名的查找或解析，例如将www表项映射到IP地址。IP地址管理的这第三项关键功能处理的是，通过使用名字而不是IP地址建立IP通信，简化了IP通信的复杂度。毕竟，被映射的IP地址必须与IP地址规划保持一致。

在本书前三部分中，讨论了组成这三项核心功能的各项技术。在第Ⅳ部分[㊀]的IPAM实践中，解释了它们的相互关系以及紧密一致地管理它们的各项实践。多数IP网络在不断地发生着变化，原因是日常的商务需求，如开新店、办公场所关闭或搬迁、注册公司以及新的设备和设备类型都需要IP地址。影响IP网络的这些变化以及其他变化对现有IP地址规划具有重大影响。随着用户数和IP地址数的增加，以及子网（或站点）数量增加，IP地址分配、各项指派以及相关DNS服务器和DHCP服务器配置等的跟踪记录以及管理任务的复杂度也增加了。

㊀ 实际上，在相应技术章节中，讨论了组成IPAM实践的几项措施，而且将这几项措施汇总于第Ⅳ部分的整体实践中。

如今执行IPAM功能的最常见方法包括使用电子表格跟踪记录IP地址，使用文本编辑器或微软Windows配置DHCP和DNS服务器。如此，将使用样例电子表格数据和配置文件范例，将其应用到名为IPAM全球公司的一个虚构组织机构，由此在整部书通篇展示IPAM概念。这样做的意图是将技术和配置细节与一个真实世界范例联系起来。

本书结构

本书分成为四部分。本书前三部分的焦点分别是三项核心IPAM功能：IP寻址和管理、DHCP和DNS。第Ⅳ部分集成这三部分，描述管理技术和实践。

第Ⅰ部分：IP寻址。本部分给出IPv4、IPv6以及IP分配和子网划分技术的详细综述。

第1章：因特网协议。本章从回顾IP首部开始，到分类的、无类的和专用IP寻址，全面讲解IP（IPv4），讨论因特网协议的演化过程以及作为保留全球IP地址空间关键技术的网络地址转换和私有寻址技术的发展过程。

第2章：IPv6（因特网协议版本6）。本章描述IPv6首部和IPv6寻址，包括地址表示、结构和当前因特网编号管理局（IANA）分配。本章包括依据类型而分的各种地址（即保留地址、全局单播地址、唯一本地单播地址、链路本地地址和组播地址）分配的详细讨论，也描述了特殊用途的地址，包括请求（solicited）节点地址和节点信息查询地址。本章接下来讨论修改的EUI-64算法和地址自动配置，最后以保留的子网任意播地址和IPv6主机所必需的地址讨论结束本章。

第3章：IP地址分配。本章讨论IPv4和IPv6地址空间IP地址块分配的各项技术，包括最佳拟合层次结构地址分配逻辑和范例，以及IPv6稀疏的和随机的分配方法。本章也讨论独特的本地地址空间和因特网注册机构的作用。地址块分配是IP地址管理的一项重要功能，它为DHCP和DNS服务的配置奠定了基础。

第Ⅱ部分：动态主机配置协议（DHCP）。本部分给出IPv4 DHCP和IPv6 DHCP的概述，并讲解依赖于DHCPv6的各项应用、DHCP服务器部署策略以及DHCP和相关的网络访问安全。

第4章：DHCP。本章描述DHCP，包括协议状态、消息格式、选项和范例的讨论。给出标准选项参数及其描述的一览表。

第5章：用于IPv6的DHCP（DHCPv6）。本章讲解DHCPv6，包括与DHCPv4的比较、消息格式、选项和范例。并给出DHCPv6选项参数的一览表。

第6章：DHCPv6的各项应用。在前面两章的技术性讨论之后，本章重点突出了DHCP的终端用户工具，描述依赖于DHCP的各项关键应用，包括VoIP设备准备工作、宽带接入准备工作、PXE（Preboot execute Environment，预启动执行环境）客户端初始化和租期限制。

第7章：DHCP服务器部署策略。本章从服务器尺寸确定、数量和位置等方面讲解DHCP服务器部署的折中考虑因素。将讨论涉及分布式方法和中心式方法的DHCP部署选项，也将讨论冗余的DHCP配置。

第8章：DHCP和网络接入安全。本章讲解DHCP安全考虑因素，并讨论DHCP

作为一个组件的网络接入安全问题。描述一个 DHCP 受控的门户配置范例，作为有关网络接入控制（NAC）方法的一个总结，其中包括基于 DHCP 的方法，基于交换机的、Cisco NAC 和微软 NAP 方法。

第Ⅲ部分：域名系统（DNS）。本部分描述 DNS 协议、DNS 应用、部署策略和相关联的配置以及安全（包括 DNS 服务器和配置的安全以及 DNSSEC）。

第 9 章：DNS 协议。本章给出 DNS 的综述，包括 DNS 概念、消息细节以及协议扩展的讨论。讲解的 DNS 概念包括基本解析过程，前向域和反向域、根信息、本地-主机域等的域树，以及解析器配置。消息细节包括 DNS 的编码（包括 DNS 头部、标签格式）以及国际域名的综述。DNS 更新消息格式化过程也作为 EDNS0 加以讨论。

第 10 章：DNS 应用和资源记录。在第 9 章的基础上，本章描述依赖于 DNS 的关键应用，包括域名解析、服务定位、ENUM（Telephone Number Mapping，电话号码映射）、采用黑/白名单列表的反垃圾技术、SPF（Sender Policy Framework，发送者策略框架）、Sender（发送者 ID）ID 和 DKIM（Domain Keys Identified Mail，域名密钥可识别的邮件）。在相关联资源记录的上下文中给出应用支持的讨论内容。

第 11 章：DNS 服务器部署策略。在本章中，讲解 DNS 部署策略和所做出的折中。DNS 服务器部署场景包括外部 DNS、因特网缓存、隐藏的主控/从属（masters/slaves）、多主控、视图、转发、内部根和任意播。

第 12 章：保障 DNS 安全（第Ⅰ部分）。本章是有关 DNS 安全的两章中的第Ⅰ部分。本章讲解与 DNS 安全有关的各种话题，不包括 DNSSEC（DNS 安全扩展）（在有关 DNSSEC 的一章中讲述）。首先给出已知的 DNS 弱点，接下来给出每个弱点的缓解方法。

第 13 章：保障 DNS 安全（第Ⅱ部分）：DNSSEC。本章详细讲解 DNSSEC，讨论产生密钥、区域（zone）签名、安全地解析名字和轮换（rolling）密钥的过程，还讨论一个范例配置。

第Ⅳ部分：IP 地址管理（IPAM）集成。本部分整合前三部分，讨论了紧密一致管理 IP 地址空间的各项技术，包括对 DHCP 和 DNS 的影响。

第 14 章：IPAM 实践。本章描述日常 IP 管理功能，包括 IP 地址分配和指派、重新编号、移动、分割、合并、DHCP 和 DNS 服务器配置、地址清单管理、故障管理、性能监测和灾难恢复。本章是围绕 FCAPS 网络管理模型组织的，强调了将一种科学性的“网络管理”方法应用到 IPAM 的必要性。

第 15 章：IPv6 部署和 IPv4 共存。在一个 IPv4 网络中实现 IPv6，将促进 IPv4 和 IPv6 协议的长期共存。本章给出共存策略的细节，将其分组为有关双栈、隧道方法和转换技术等节。涵盖内容包括 6to4、ISATAP、6over4、Teredo、DSTM，及隧道代理打隧道方法，和 NAPT-PT、SOCKS、TRT、ALG 以及栈中打楔子或 API 转换方法。本章以一些基本迁移场景结束。

Timothy Rooney
宾夕法尼亚州，Norristown

读者需求调查表

个人信息

姓名：		出生年月：		学历：	
联系电话：		手机：		E-mail：	
工作单位：				职务：	
通讯地址：				邮编：	

1. 您感兴趣的科技类图书有哪些？

□自动化技术　□电工技术　□电力技术　□电子技术　□仪器仪表　□建筑电气

□其他（　　）以上各大类中您最关心的细分技术（如 PLC）是：（　　　）

2. 您关注的图书类型有：

□技术手册　□产品手册　□基础入门　□产品应用　□产品设计　□维修维护

□技能培训　□技能技巧　□识图读图　□技术原理　□实操　□应用软件

□其他（　　）

3. 您最喜欢的图书叙述形式为：

□问答型　□论述型　□实例型　□图文对照　□图表　□其他（　　）

4. 您最喜欢的图书开本为：

□口袋本　□32 开　□B5　□16 开　□图册　□其他（　　）

5. 您常用的图书信息获得渠道为：

□图书征订单　□图书目录　□书店查询　□书店广告　□网络书店　□专业网站

□专业杂专　□专业报纸　□专业会议　□朋友介绍　□其他（　　）

6. 您常用的购书途径为：

□书店　□网络　□出版社　□单位集中采购　□其他（　　）

7. 您认为图书的合理价位是（元/册）：

手册（　　）图册（　　）技术应用（　　）技能培训（　　）基础入门（　　）其他（　　）

8. 您每年的购书费用为：

□100 元以下　□101～200 元　□201～300 元　□300 元以上

9. 您是否有本专业的写作计划？

□否　□是（具体情况：　　）

非常感谢您对我们的支持，如果您还有什么问题欢迎和我们联系沟通！

地址：北京市西城区百万庄大街 22 号　机械工业出版社电工电子分社　邮编：100037

联系人：张俊红　联系电话：13520543780　传真：010-68326336

电子邮箱：buptzjh@163.com（可来信索取本表电子版）

编著图书推荐表

姓名		出生年月		职称/职务		专业	
单位				E-mail			
通讯地址						邮政编码	
联系电话		研究方向及教学科目					
个人简历(毕业院校、专业、从事过的以及正在从事的项目、发表过的论文)							
您近期的写作计划有:							
您认为目前市场上最缺乏的图书及类型有:							

地址:北京市西城区百万庄大街22号　机械工业出版社,电工电子分社

邮编:100037　网址:www.cmpbook.com

联系人:张俊红　电话:13520543780/010-88379768　010-68326336(传真)

E-mail:buptzjh@163.com(可来信索取本表电子版)

目　录

译者序
原书前言

第Ⅰ部分　IP寻址

第1章　因特网协议 …… 1
1.1　因特网协议历史的精彩部分 …… 1
1.1.1　IP首部 …… 3
1.2　IP寻址 …… 4
1.2.1　基于类别的寻址 …… 5
1.2.2　因特网增长带来的痛苦 …… 6
1.2.3　私有地址空间 …… 8
1.3　无类别寻址 …… 10
1.4　特殊用途地址 …… 11
第2章　IPv6（因特网协议版本6） …… 12
2.1　引言 …… 12
2.1.1　IPv6关键功能特征 …… 13
2.1.2　IPv6首部 …… 13
2.1.3　IPv6寻址 …… 14
2.1.4　地址表示法 …… 15
2.1.5　地址结构 …… 16
2.2　IPv6地址分配 …… 17
2.2.1　::/3——保留地址 …… 18
2.2.2　2000::/3——全局单播地址空间 …… 18
2.2.3　FC00::/7——唯一本地地址空间 …… 18
2.2.4　FE80::/10——链路本地地址空间 …… 19
2.2.5　FF00::/8——组播地址空间 …… 19
2.2.6　特殊情形的组播地址 …… 22
2.2.7　带有内嵌IPv4地址的IPv6地址 …… 24
2.3　IPv6地址自动配置 …… 24
2.4　邻居发现 …… 25
2.4.1　改进的EUI-64接口标识符 …… 25
2.4.2　重复地址检测 …… 26

2.5 保留的子网任意播地址 …… 27
2.6 必备的主机 IPv6 地址 …… 28
第 3 章 IP 地址分配 …… 29
3.1 地址分配逻辑 …… 31
3.1.1 顶层分配逻辑 …… 32
3.1.2 第二层分配逻辑 …… 33
3.1.3 地址分配第 3 部分 …… 37
3.1.4 分配均衡和跟踪 …… 38
3.1.5 IPAM 全球公司的公开地址空间 …… 40
3.2 IPv6 地址分配 …… 41
3.2.1 最佳拟合分配 …… 41
3.2.2 稀疏分配方法 …… 42
3.2.3 随机分配 …… 43
3.2.4 唯一的本地地址空间 …… 44
3.3 IPAM 全球公司的 IPv6 地址分配 …… 44
3.4 因特网注册机构 …… 48
3.4.1 RIR 地址分配 …… 50
3.4.2 地址分配效率 …… 52
3.5 多穴接入法和 IP 地址空间 …… 52
3.6 地址块分配和 IP 地址管理 …… 54

第Ⅱ部分 动态主机配置协议（DHCP）

第 4 章 DHCP …… 55
4.1 引言 …… 55
4.2 DHCP 综述 …… 55
4.2.1 DHCP 消息类型 …… 58
4.2.2 DHCP 报文格式 …… 60
4.3 DHCP 服务器和地址指派 …… 62
4.3.1 依据类的设备识别 …… 63
4.4 DHCP 选项 …… 65
4.5 动态地址指派的其他方式 …… 75
第 5 章 用于 IPv6 的 DHCP（DHCPv6） …… 76
5.1 DHCP 比较：DHCPv4 和 DHCPv6 …… 76
5.2 DHCPv6 地址指派 …… 77
5.3 DHCPv6 前缀委派 …… 79
5.4 DHCPv6 对地址自动配置的支持 …… 79
5.4.1 DHCPv6 消息类型 …… 79

5.4.2 DHCPv6 报文格式 …… 81
5.5 设备唯一标识符 …… 82
5.5.1 DUID-LLT …… 82
5.5.2 DUID-EN …… 83
5.5.3 DUID-LL …… 83
5.6 身份关联 …… 83
5.7 DHCPv6 选项 …… 84
第6章 DHCPv6 的各项应用 …… 91
6.1 多媒体设备类型特定配置 …… 91
6.2 宽带订户配置信息准备 …… 92
6.3 有关租期指派或限制的各项应用 …… 96
6.4 预启动执行环境客户端 …… 96
6.4.1 PPP/RADIUS 环境 …… 97
6.4.2 移动 IP …… 98
第7章 DHCP 服务器部署策略 …… 99
7.1 DHCP 服务器平台 …… 99
7.1.1 DHCP 软件 …… 99
7.1.2 虚拟机 DHCP 部署 …… 99
7.1.3 DHCP 仪器设备 …… 99
7.2 中心式 DHCP 服务器部署 …… 100
7.3 分布式 DHCP 服务器部署 …… 101
7.4 服务器部署设计考虑 …… 102
7.5 在边缘设备上部署 DHCP …… 105
第8章 DHCP 和网络接入安全 …… 107
8.1 网络接入控制 …… 107
8.1.1 采用 DHCP 的区分性地址指派 …… 107
8.2 其他接入控制方法 …… 112
8.2.1 DHCP LeaseQuery …… 112
8.2.2 层2交换机提醒 …… 112
8.2.3 802.1X …… 113
8.2.4 Cisco 网络接纳控制 …… 114
8.2.5 微软网络接入保护 …… 114
8.3 使 DHCP 安全 …… 116
8.3.1 DHCP 威胁 …… 116
8.3.2 DHCP 威胁缓解措施 …… 117
8.3.3 DHCP 认证 …… 117

第Ⅲ部分　域名系统（DNS）

第9章　DNS协议 …… 119
9.1　DNS综述——域和解析 …… 119
9.1.1　域层次结构 …… 119
9.2　名字解析 …… 120
9.3　区域和域 …… 123
9.3.1　区域信息的传播 …… 125
9.3.2　反向域 …… 126
9.3.3　IPv6反向域 …… 129
9.3.4　其他区域 …… 132
9.4　解析器配置 …… 132
9.5　DNS消息格式 …… 134
9.5.1　域名的编码 …… 134
9.5.2　名字压缩 …… 135
9.5.3　国际域名 …… 136
9.5.4　DNS消息格式 …… 137
9.5.5　DNS更新消息 …… 143
9.5.6　DNS扩展（EDNS0） …… 145
9.5.7　资源记录 …… 146
第10章　DNS应用和资源记录 …… 147
10.1　引言 …… 147
10.1.1　资源记录格式 …… 147
10.2　名字-地址查询应用 …… 149
10.2.1　主机名和IP地址解析 …… 149
10.2.2　别名主机和域名解析 …… 150
10.2.3　网络服务定位 …… 151
10.2.4　主机和文本信息查找 …… 152
10.2.5　DNS协议运营性的记录类型 …… 154
10.2.6　动态DNS更新唯一性验证 …… 155
10.2.7　电话号码解析 …… 156
10.3　EMAIL和反垃圾邮件管理 …… 159
10.3.1　电子邮件和DNS …… 159
10.3.2　白名单或黑名单方法 …… 163
10.3.3　发送者策略框架 …… 163
10.3.4　发送者ID …… 167
10.3.5　域名密钥可识别的邮件 …… 168

10.3.6 历史上出现过的电子邮件资源记录类型 …… 170
10.4 安全应用 …… 171
10.4.1 保障名字解析的安全——DNSSEC 资源记录类型 …… 171
10.4.2 其他面向安全的 DNS 资源记录类型 …… 176
10.4.3 地理定位查找 …… 179
10.4.4 非 IP 主机地址查找 …… 180
10.4.5 Null 记录类型 …… 181
10.5 试验型的名字-地址查找记录 …… 181
10.5.1 IPv6 地址链——A6 记录（试验型的） …… 181
10.5.2 APL——地址前缀列表记录（试验型的） …… 182
10.6 资源记录小结 …… 183
第 11 章 DNS 服务器部署策略 …… 187
11.1 通用的部署指导原则 …… 187
11.2 通用的部署构造块 …… 188
11.3 外部-外部分类 …… 189
11.3.1 外部 DNS 服务器 …… 190
11.4 外部-内部分类 …… 193
11.4.1 外部网 DNS 服务器部署 …… 193
11.5 内部-内部分类 …… 194
11.5.1 内部解析 DNS 服务器 …… 194
11.5.2 内部委派 DNS 主/从服务器 …… 195
11.5.3 内部根服务器 …… 196
11.5.4 隐秘的从属 DNS 服务器 …… 198
11.5.5 多层服务器配置 …… 198
11.6 内部-外部分类 …… 199
11.6.1 混合权威/缓存 DNS 服务器 …… 199
11.6.2 专用的缓存服务器 …… 200
11.6.3 外网解析服务器 …… 203
11.7 交叉角色分类 …… 204
11.7.1 分割视图 DNS 服务器 …… 204
11.7.2 采用任意播地址部署 DNS 服务器 …… 209
11.8 将所有情况整合起来 …… 212
第 12 章 保障 DNS 安全（上） …… 213
12.1 DNS 弱点 …… 213
12.1.1 解析攻击 …… 214
12.1.2 配置攻击和服务器攻击 …… 215
12.1.3 拒绝服务攻击 …… 216

12.2 缓解方法 …… 216
12.3 非DNSSEC安全记录 …… 217
12.3.1 TSIG——事务签名记录 …… 217
12.3.2 SIG (0)——涵盖空类型的签名记录 …… 218
12.3.3 KEY——密钥记录 …… 219
12.3.4 TKEY——事务密钥记录 …… 219
第13章 保障DNS安全（下）：DNSSEC …… 221
13.1 数字签名 …… 221
13.2 DNSSEC综述 …… 222
13.3 配置DNSSEC …… 224
13.3.1 产生密钥 …… 224
13.3.2 将密钥添加到区域文件 …… 230
13.3.3 对区域签名 …… 230
13.3.4 链接信任链 …… 242
13.4 DNSSEC解析过程 …… 245
13.4.1 验证签名 …… 245
13.4.2 经过认证的存在性拒绝 …… 247
13.4.3 在一个信任链中的父区域委派 …… 248
13.5 密钥轮换 …… 250
13.5.1 自动的信任锚点轮换 …… 252
13.5.2 DNSSEC和动态更新 …… 254
13.5.3 DNSSEC部署考虑 …… 254

第Ⅳ部分 IP地址管理（IPAM）集成

第14章 IPAM实践 …… 256
14.1 FCAPS概述 …… 256
14.2 共同的IP管理任务 …… 257
14.3 配置管理 …… 257
14.3.1 地址分配任务 …… 258
14.3.2 地址删除任务 …… 263
14.3.3 地址重新编号或移动任务 …… 264
14.3.4 地址块/子网分割 …… 267
14.3.5 地址块/子网合并 …… 268
14.3.6 DHCP服务器配置 …… 268
14.3.7 DNS服务器配置 …… 269
14.3.8 服务器升级管理 …… 272
14.4 故障管理 …… 272

14.4.1 故障检测 …… 272
14.4.2 排错和故障解决 …… 273
14.5 记账管理 …… 280
14.5.1 确保清单准确 …… 280
14.5.2 地址回收 …… 283
14.6 性能管理 …… 284
14.6.1 服务监测 …… 284
14.6.2 地址容量管理 …… 284
14.6.3 审计和报告 …… 285
14.7 安全管理 …… 286
14.8 灾难恢复/商务持续性 …… 286
14.9 ITIL 过程映射 …… 287
14.9.1 ITIL 过程域 …… 287
14.10 结论 …… 291
第 15 章 IPv6 部署和 IPv4 共存 …… 292
15.1 引言 …… 292
15.1.1 为什么要实现 IPv6 …… 293
15.1.2 IPv4-IPv6 共存技术 …… 293
15.2 双栈方法 …… 294
15.2.1 部署 …… 294
15.2.2 DNS 考虑 …… 295
15.2.3 DHCP 考虑 …… 296
15.3 打隧道方法 …… 296
15.3.1 IPv4 网络上传输 IPv6 报文的打隧道场景 …… 297
15.3.2 隧道类型 …… 299
15.3.3 IPv6 网络上传输 IPv4 报文的打隧道场景 …… 308
15.3.4 隧道法总结 …… 309
15.4 转换方法 …… 310
15.4.1 无状态 IP/ICMP 转换算法 …… 310
15.4.2 协议栈中的肿块 …… 311
15.4.3 API 中的肿块 …… 312
15.4.4 带有协议转换的网络地址转换（NAT-PT）——被废弃不用 …… 313
15.4.5 带有协议转换的网络地址端口转换（NAPT-PT）——废弃不用 …… 313
15.4.6 SOCKS IPv6/IPv4 网关 …… 313
15.4.7 传输中继转换器 …… 314
15.4.8 应用层网关 …… 314

15.5 应用迁移 …… 315
15.6 规划 IPv6 部署过程 …… 315
15.6.1 服务提供商部署选项 …… 315
15.6.2 企业部署场景 …… 317
15.6.3 核心网络迁移场景 …… 318
15.6.4 服务器侧迁移 …… 318
15.6.5 客户端侧迁移 …… 320
15.6.6 客户端-服务器迁移 …… 320
15.6.7 总体 IPv6 实现规划 …… 321
参考文献 …… 323
词汇表 …… 332
RFC 索引 …… 333

第I部分 IP寻址

第Ⅰ部分开始讨论 IPAM 的第一块基石：IP 寻址。本部分涵盖 IPv4 协议、IPv6 协议和地址块管理技术。

第1章 因特网协议

1.1 因特网协议历史的精彩部分

因特网协议（IP）改变了所有事物。我在美国电话电报公司（AT&T）贝尔实验室的早期生涯，当时是20世纪80年代中期，我们使用哑终端连接到大型计算机的主机（mainframe），连网领域才开始支持智能的分布式处理，这个处理过程从一个中心化的大型计算机主机到连网的服务器、路由器，最终才连接到个人计算机。既然提到了我的过去，就继续说，在稍后一段时间，许多竞争性的连网技术都希望竞争得到在企业部署的位置，但却没有分出伯仲。完全不同的连网协议和技术的部署，导致各组织机构间不能相互通信，直到20世纪90年代，多亏了因特网的广受拥戴，因特网协议才成为世界范围事实上的连网协议。

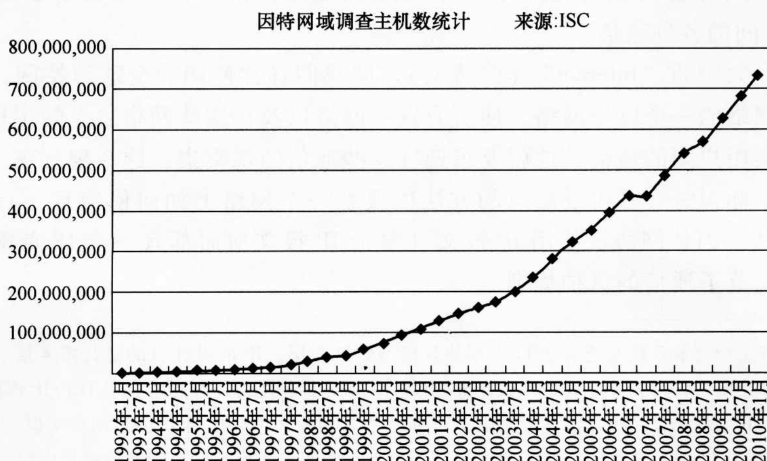


图1-1 1993-2010[3]期间因特网主机的增长情况（来源：ISC）

如今的现状是，因特网协议是世界范围内得到最广泛部署的网络层㊀协议。从20世纪60年代开始的美国国防部资助的一项连网项目诞生伊始，到目前为止，传输控制协议/因特网协议（TCP/IP）族已经演化并规模扩展到可支持数百台计算机到数亿台计算机构成的各种类型网络。事实上，依据因特网系统联盟（Internet Systems Consortium，ISC）调查结果，在因特网上设备或主机㊁的数量在2010年初就超过7亿3000万台，在过去6年间，每年平均增加7500万台主机（见图1-1）。因特网已经从一个研究项目无缝地扩展到连接有超过7亿3000万台主机的一个巨型网络，这样的事实是对因特网开发人员的理想憧憬的实证，以及是该网络底层技术设计强健性的实证。

因特网协议“最初”是1980年在请求评述（RFC㊂）760[1]和791[2]中定义的，是由德高望重的Jon Postel编辑的。我们引用“最初”的说法，是因为Postel先生在他的前言中指出，虽然在RFC 791中它被称作版本4（IPv4），但它是构筑在ARPA（高级研究项目局，这是美国国防部的一个局）因特网协议前6个较早版本基础上的。RFC 791陈述说，因特网协议执行两项基本功能：寻址和分段。虽然这看起来可能使那时及以后实现的因特网协议的许多其他功能和特征不那么重要，但对于任何一名协议设计者而言，它实际上重点突出了这两个主要话题的重要性。分段实施如下功能：将消息分割成许多IP报文，从而使它们可在具有有效报文尺寸约束的网络上进行传输，并在目的地以正确的顺序对报文实施组装。当然，寻址是本书的关键话题之一，所以确保要求可达性的主机的唯一可寻址能力，对于基本协议操作而言，这就是至关重要的了。

因特网已经成为日常个人和商务工作的一个不可分割的工具，其上有各种应用，如电子邮件、社会网络、万维网浏览、无线接入和语音通信。因特网确实已经成为现代社会的一项关键组成元素。正如你所关注的，“因特网”（Internet）这个术语从因特网技术早期开发人员所用术语的小写形式变化而来的，现在指互相连接网络或“互联网”间的各种通信。

如今，大写的“Internet”（因特网），即我们日常使用的全球因特网，已经成为互相连接网络的一个巨型网络。使所有这些网络以及在这些网络之上的主机高效地协作，并交换用户间的通信，这就要求遵守这些通信的规则集。这个规则集，即这个协议，定义了标识每台主机或端点的方法以及在一个网络上如何使信息从点A传递到点B的方法。因特网协议使用IP报文（每个IP报文前面都有一个IP首部）这个运输工具，规范了通信的这种规则。

㊀ 网络层指开放系统互连（OSI）7层协议模型的第3层。IP的设计目的是与第4层（传输层）的传输控制协议（TCP）或用户数据报协议（UDP）一起使用，因此才有了TCP/IP族的术语。在名为《IP地址管理绪论》（参考文献11）的书中，讨论了OSI模型和IP连网的常识。

㊁ “主机”（host）这个术语是相对于一台路由器或中间设备而言的，指通信路径中的一个端节点。

㊂ 因特网协议继续演化发展，它的规范是以顺序编号的RFC形式归档的。因特网工程任务组（IETF）是一个开放的没有正式成员的社团组织，它负责发布RFC。

1.1.1　IP 首部

TCP/IP 族内部的 IP 层从 TCP 或 UDP 传输层接收数据，并为该数据添加一个 IP 首部。分布于到最终目的地的整条路径上的各台路由器分析这个 IP 首部，最终将每条 IP 报文交付到该报文的最终目的地，最终目的地是由首部中的目的 IP 地址标识的。RFC 791 将 IP 地址结构定义为 32bit 结构，该结构开始是一个网络号，接下来是一个本地地址。在每个 IP 报文的首部中携带该地址。图 1-2 展示出 IP 首部的各字段。每个 IP 报文包含一个 IP 首部，接下来是报文内部的数据内容，该内容包括较高层协议的控制信息。

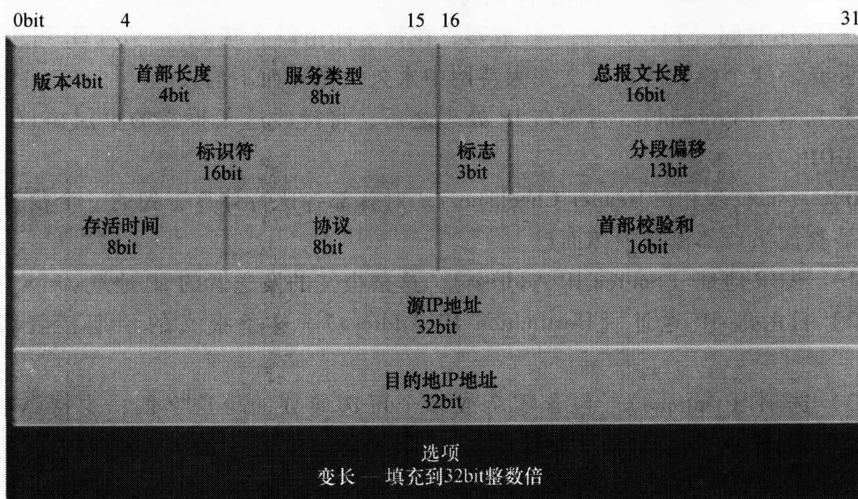


图 1-2　IPv4 首部字段[1]

（1）版本（Version）。因特网协议版本，在这种情形中是 4。

（2）首部长度（Header Length）（因特网首部长度，IHL）。以称为“字”的 32bit 单元表示的 IP 首部长度。例如，最小首部长度是 5，在图 1-2 中突出显示为浅色阴影字段，它由 5 个字 × 32bit/字 = 160bit 组成。

（3）服务类型（Type of Service）。与报文的服务质量（QoS）有关的参数。最初定义为 ToS（服务类型），这个字段由一个 3bit 优先级字段和另一个 3bit 组成，前 3bit 支持区分一个特定报文的相对重要性，后 3bit 分别用来请求低延迟、高吞吐量或高可靠性的服务。

在 RFC 2474“IPv4 和 IPv6 首部中，区分服务字段（DS 字段）的定义”（177）重新定义了原始的 ToS 字段。DS 字段（或称区分服务字段）提供了一个 6bit 的码点（DSCP，区分服务码点）字段，剩余 2bit 未用。码点映射到一项预定义的服务，接下来该服务被关联到由网络提供的一个服务等级。随着因特网权威机构以相应的服务处理法定义新的码点，则 IP 路由器可实施对应于所定义码点的路由处理方法，例如，针对延迟敏感的应用实施较高优先级的处理。

(4) 总报文长度 (Total Packet Length)。以字节 (Byte) (8bit 长的字符) 表示的整个 IP 报文的长度。

(5) 标识符 (Identification)。为每条报文赋予值,有助于在接收端对报文分段进行组装。

(6) 标志 (Flags)。这个 3bit 字段定义如下:

1) bit 0 是保留的,必须为 0。

2) bit 1——不要分段——表明这条报文不能分段。

3) bit 2——更多分段——表明这条报文是一个分段,但这不是最后一个分段。

(7) 分段偏移 (Fragment Offset)。标识这个分段相对于原始报文开始部分的位置,是以 64bit 的"双字"为单位标识的。

(8) 存活时间 (TTL)。一个计数器,在每个路由跳要减 1;一旦 TTL 到达 0,则报文被丢弃。这个参数防止报文在因特网中永久地循环而不消失。

(9) 协议 (Protocol)。指明在 IP 处理之后,将接收这条报文的上层协议,例如 TCP 或 UDP。

(10) 首部校验和 (Header Checksum)。对首部各比特计算得到的一个校验和值,目的仅是验证该首部没有损坏而已。

(11) 源 IP 地址 (Source IP Address)。这条报文的发送者的 IP 地址。

(12) 目的地 IP 地址 (Destination IP Address)。这条报文的预期接收者的 IP 地址。

(13) 选项 (Options)。包含零个或多个可选参数的可选字段,支持路由控制 (源路由)、诊断 (trace route (跟踪路由)、最大传输单元 (MTU) 发现) 等。

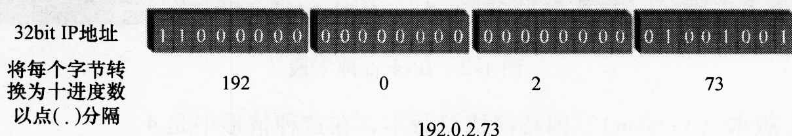


图 1-3 二进制到点分十进制的转换

如果你发现 IP 首部细节的这个解释有点滑稽,那也没有什么。现在让我们将关注焦点放在源 IP 地址和目的地 IP 地址字段以及 IP 寻址结构上。

1.2 IP 寻址

IP 地址字段是由 32bit 组成的。人们所熟悉的一个 IP 地址点分十进制表示法,反映了将 32bit 地址分割成四个 8bit 字节的想法。将四个字节的每个字节都转换为十进制,之后以十进制点或"点"(dots) 将它们隔离开。相比于将这些 32bit 作为一个大数计算的方法,这当然要容易得多!考虑图 1-3 中的 32bit IP 地址。简单地将这个地址分割成四个字节,把每个字节转换为十进制,之后将每个字节的十进制表示以"点"隔开。由此,得到"点分十进制"的术语。

1.2.1　基于类别的寻址㊀

RFC 791[2]定义了三个类别的地址：A、B 和 C 类。这些类别由图 1-4 所示的 32bit 地址的前几个比特加以识别。每个类别对应于一个特定尺寸（大小）的网络号和本地地址字段。本地地址字段可被指派到独立的主机或进一步分成子网和主机字段，我们将在后面讨论。

将地址空间分成类别的做法，为针对不同用户的需要而方便地定义不同大小的网络提供了一种方法。当时，因特网是由某些美国政府部门、大学和一些研究机构组成的。它还没有盛行成为像今天一样事实上的世界范围骨干网络，所以地址容量看来似乎是无限的。在这些字节边界上将地址空间分成类别的另一个原因是，比较容易实现网络路由。简单地检查目的地址的前几个比特，路由器就能够识别网络号字段的长度。之后路由器将简单地在它们的路由表中查找这个 IP 地址的网络号部分，并据此路由每条报文。在那些日子里，计算能力是相当有限的，所以最小化处理需求是另一项考虑因素。分类寻址的一个副作用是简单的可达性。每个点分十进制数表示二进制中的一个字节。就如我们以后将了解到的，当讨论无类别寻址时，如今的典型情况并不是上面提到的情况。

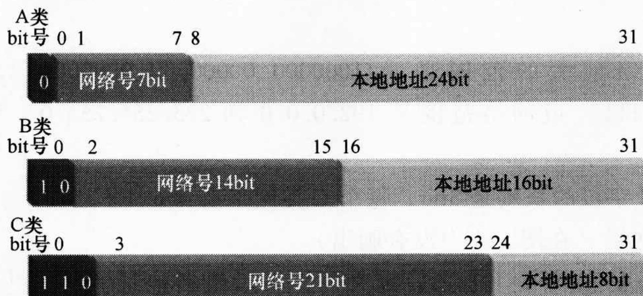


图 1-4　基于类别的寻址

仔细研究这个基于类别的寻址结构，我们可观察到一些关键点：

(1) A 类网络

■ A 类前缀以二进制 0（$[0]_2$）㊁开始，加上 7 个附加比特，或说网络地址部分总共有 8bit

■全 0 网络地址是无效的㊂。

■网络地址 $[01111111]_2 = 127$ 是一个保留地址。地址 127.0.0.1 用于一个接口上的“回环地址”。

㊀　本章后面大部分内容利用了参考文献［11］第 2 章的资料。

㊁　在可能存在二义性的场合，为了区分一个二进制 0（1bit）和一个十进制 0（7 或 8bit），我们以适当的基数作为下标。不要担忧；我们不会离题到化学方面去讨论以 O_2 表示氧气分子式，这里是简单的“以 2 为基的数字 0”。

㊂　但一些协议（例如 DHCP）使用全 0 地址作为“这个（本）”地址的位置保留符

■这样的结果是，一个 A 类网络前缀第一个字节的范围是 $[00000001]_2$ 到 $[01111110]_2 = 1 \sim 126$。

■本地地址字段的长度是 24bit。这等于每个网络地址空间（A 类网络）有多达 $2^{24} = 16777216$ 个可能的本地地址。一般来说，全 0 本地地址表示该“网络”地址，全 1 地址是网络广播地址，所以通常我们从本地地址容量中减去这两个地址，这样得到每个 A 类网络有 16777214 个主机地址。因此，10.0.0.0 是 10.0.0.0/8 的网络地址，10.255.255.255 是到 10.0.0.0/8 网络上所有主机的广播地址。

（2）B 类网络

■B 类网络以 $[10]_2$ 开始，加上 14 个附加比特，或说网络地址部分总共有 16bit。

■以二进制表示的 B 类网络前缀范围是 $[10000000\ 00000000]_2$ 到 $[10111111\ 11111111]_2$ 或网络范围是 128.0.0.0 到 191.255.0.0，得到 16384 个网络地址。

■本地地址字段的长度是 16bit，每个 B 类网络有 65536-2 = 65534 个可能的主机地址。

（3）C 类网络

■C 类网络以 $[110]_2$ 开始，加上 21 个比特附加，或说网络地址部分总共有 24bit。

■C 类网络前缀的范围是 $[11000000\ 00000000\ 00000000]_2$ 到 $[11011111\ 11111111\ 11111111]_2$ 或网络范围是 192.0.0.0 到 223.255.255.0，得到 2097152 个网络。

■本地地址字段的长度是 8bit，每个 C 类网络有 256-2 = 254 个可能的主机地址。

（4）D 类网络（在图 1-4 中没有画出）

■D 类网络是在 RFC 791 之后定义的，表示组播地址，是从 $[1110]_2$ 开始的。组播可用于流化应用，在其中多个用户或署名用户（subscribers）从一个共同的源接收一系列的 IP 报文。换句话说，具有一个共同组播地址的多台主机将接收到发送到组播组或地址的所有 IP 流量。组播网络没有网络部分和主机部分，原因是一个组播组的各成员可能处于许多不同的物理网络之上。

■D 类网络的范围是从 $[11100000\ 00000000\ 00000000\ 00000000]_2$ 到 $[11101111\ 11111111\ 11111111\ 11111111]_2$ 或网络范围是 224.0.0.0 到 239.255.255.255，得到 268435456 个组播地址。

（5）E 类网络（在图 1-4 中没有画出）

■保留以 $[1111]_2$（E 类）开始的网络。

1.2.2 因特网增长带来的痛苦

因为拥有似乎无限的 IP 地址容量，至少在整个 20 世纪 80 年代看来是这样的，所以 A 类和 B 类网络就通常分配给请求这些网络地址的单位或组织。之后接收到网络地址的组织机构将他们的 A 类或 B 类网络沿字节边界在他们的组织内部进行切分

或子网划分[⊖]。要记住的是，每个“网络”，即使在一个公司内部，也需要有一个唯一的网络号或前缀来维持地址唯一性，并维持路由完整性。

子网划分法为通信和路由协议更新提供了路由边界。IP报文要穿越的每个网络，都要求有其自己的IP网络号（网络地址）。随着越来越多的公司通过请求IP地址空间，寻求加入到因特网，因特网注册机构（Internet Registries，该机构负责分配IP地址空间）被强迫放慢分配地址的速度。那些从因特网注册机构请求IP地址的公司，很快面临着日渐紧迫（Stringent）的应用需求，因此被授权使用所请求地址空间的一部分（注：也许是几分之一）。在不得已以较小网络地址块分配应对地址需求的情况下，许多机构被迫在非字节边界进行子网划分。

无论是否在字节边界实施操作，通过和网络地址一起指派一个网络掩码的手段，子网划分法得以顺利进行。网络掩码是一个整数，代表以比特表示的网络前缀的长度。有时这也被称作掩码长度。例如，一个A类网络的掩码长度为8，一个B类网络的掩码长度为16，一个C类网络的掩码长度为24。采用本质上扩展网络号长度（路由器在每条报文中均要实施网络号长度的检查）的方法，就能够支持较大数量的网络，并可更灵活地分配地址空间。这在图1-5中进行了图示说明。

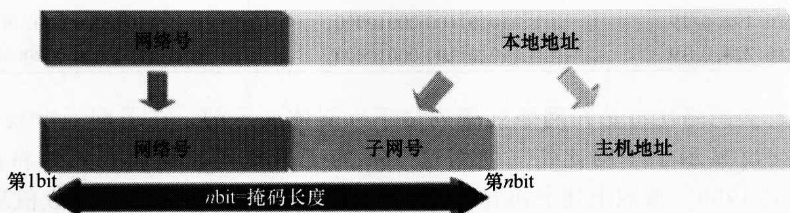


图1-5 子网划分法采用每个网络较少主机的方法，提供了更多的“网络”

路由器需要以这个掩码长度为它们所服务的每个子网进行配置。这就使这些路由器能够对IP地址实施“掩码”，例如，仅输出所指明的网络，并在32bit IP地址内部对各比特进行子网划分，从而在不依赖于地址类别的条件下，就能够进行高效的路由操作。依据这个扩展的网络号，该路由器能够相应地对报文进行路由。

网络地址和掩码长度最早是以点分十进制表示法指定32bit掩码的做法加以表示的。这种表示法是如下得到的：将一个32bit数的前$n$bit分别设置为1，剩下的$32-n$bit分别设置为0，之后将得到的32bit转换为点分十进制。

例如，为表示一个19bit长的网络掩码，可做如下步骤操作：

1）产生32bit数，有19个1和13个0：11111111 11111111 11100000 00000000。

2）将之分隔成字节：11111111. 11111111. 11100000. 00000000。

3）转换为点分十进制数：255. 255. 224. 0。

例如，采用这个19bit掩码的172. 16. 168. 0网络表示为172. 16. 168. 0/255. 255. 224. 0。

谢天谢地，这种方法被一种比较简单的表示法所替代：现在带有网络地址的掩码

⊖ “子网划分”（subnet）术语在本章上下文中频繁地被用作一个动词，指产生一个子网的动作或行为。

表示为〈网络地址〉/〈掩码长度〉。虽然这种表示法阅读理解起来比较容易，但这并不能使我们省掉等价的二进制练习！例如，B 类网络 172.16.0.0 将被表示为 172.16.0.0/16。“/16”（斜杠 16）指明前 16bit，在这种情形中是前两个字节，表示网络前缀。

下面是这个网络的二进制表示：

网络地址	网络前缀	本地地址
172.16.0.0/16	10101100 00010000	00000000 00000000

我们使用一个 19bit 的掩码，对这个网络进行子网划分。将点分十进制表示扩展成二进制表示：

网络地址	网络前缀	子网本地地址
172.16.0.0/19	10101100 00010000	**000**00000 00000000
172.16.32.0/19	10101100 00010000	**001**00000 00000000
172.16.64.0/19	10101100 00010000	**010**00000 00000000
172.16.96.0/19	10101100 00010000	**011**00000 00000000
172.16.128.0/19	10101100 00010000	**100**00000 00000000
172.16.160.0/19	10101100 00010000	**101**00000 00000000
172.16.192.0/19	10101100 00010000	**110**00000 00000000
172.16.224.0/19	10101100 00010000	**111**00000 00000000

注意 B 类网络比特是在网络前缀列之下以斜体表示的，在子网列中我们以较粗的黑斜体突出显示了子网比特。使用这 3bit 的子网掩码，我们有效地将网络号从 16bit 扩展到 19bit。按照上述突出显示的子网比特，将这 3bit 的二进制数值从 $[000]_2$ 增加到 $[111]_2$，这样，我们就可使用这 3bit 的子网掩码扩展，得到 $2^3=8$ 个子网。之后，可对路由器进行路由配置，使用前 19bit 识别地址的网络部分，方法是对服务相应掩码长度子网的路由器进行配置，例如 172.16.128.0/19，之后使路由器通过路由协议传播到这个网络的可达性信息。称为可变长度子网掩码（Variable Length Subnet Masking，VLSM）的这种技术正在日渐变成流行的常用方法，它有助于在一个组织机构内部将尽可能多的 IP 地址容量从指派的地址空间中释放出来。

在 IP 存在的前一二十年中，两层网络/子网模型运行得很好。但是，在 20 世纪 90 年代早期，对 IP 地址的需求持续地迅猛增长，有越来越多的公司期望得到 IP 地址空间，以发布网站。如果按照当时的使用速率，预计在世纪之交之前地址空间就会消耗完！因特网使用指导组织，即因特网工程任务组（IETF）果断地实施了两项关键策略来扩展 IP 地址的可使用寿命，即支持私有地址空间［终版 RFC 1918（7）］和无类域间路由［CIDR，RFC 1517-1519（参考文献［4-6］）］。在这个时间段，IETF 也开始对拥有巨大地址空间的 IP 新版本展开工作，即 IPv6（IP 版本 6），对此我们将在下一章进行讨论。

1.2.3　私有地址空间

回顾一下我们的论断，即为了维持地址唯一性和路由完整性，在一个组织机构内

部的每个“网络”都需要有一个唯一的网络号或前缀。随着越来越多的组织机构连接到因特网上，因特网成为黑客们渗透组织机构网络的一个极可能被利用的工具。许多机构实施防火墙，基于有关 IP 首部值的特定准则来过滤 IP 报文，例如源地址或目的地址、UDP 或 TCP 以及其他信息。这就防护了“内部”地址空间和“外部”地址空间之间 IP 地址空间的分隔，这种做法与 IETF 内部的地址预留工作非常好地配合起来。

IETF 发布了多个 RFC 修订稿，使 RFC 1918 成为一个标准文档，它将如下网络地址集合定义为“私有的”。

1）10.0.0.0 ~ 10.255.255.255（10/8 网络）——等价于 1 个 A 类。

2）172.16.0.0 ~ 172.31.255.255（172.16/12 网络）——等价于 16 个 B 类。

3）192.168.0.0 ~ 192.168.255.255（192.168/16 网络）——等价于 1 个 B 类或 256 个 C 类。

“私有的”这个术语意味着这些地址在因特网上是不可路由的。但是，在一个组织机构内部，这些地址可被用于在内部网络上路由 IP 流量。因此，我的笔记本计算机被分配一个私有 IP 地址，我能够向我的同事们发送电子邮件，他们也有私有地址。本质上而言，我所在的机构定义了一个私有因特网，有时该网络被称为一个内联网。位于我所在机构内部的路由器可配置成：在所分配私有 IP 网络间实施路由，在这些网络间的 IP 流量永不会到达因特网[⊖]。

因为我正在使用一个私有 IP 地址，在本机构外部的某个人，他在防火墙之外，是不能直接到达我的笔记本计算机的。处于外部网络的任何人，他发送在 IP 首部以我的私有地址作为目的地址的报文，该报文都将不能到达我的计算机，原因是因特网路由器不会路由这些报文。但是，如果我希望在外部通过因特网发起一条连接，目的是查验我在股票市场上将损失多少钱时，该怎么办呢？对于要求接入到因特网的雇员而言，普遍采用具有网络地址转换（NAT）功能的防火墙来将一个企业用户的私有 IP 地址转换为一个公开的或可路由的 IP 地址（该地址是从企业的公开地址空间取得的）。

典型的 NAT 设备提供地址池功能，它将相对少量的公开可路由（非私有的）IP 地址放在地址池中，以动态方式为偶尔访问因特网的大量雇员所使用。NAT 设备将两条 IP 连接桥接在一起：内部到 NAT 设备的通信使用私有地址空间，而 NAT 设备到因特网的通信使用公开 IP 地址。NAT 设备负责跟踪记录内部雇员地址到外部使用的公开地址之间的映射关系。

这在图 1-6 中进行了形象展示，其中内部网络使用 10/8 地址空间，外部或公开寻址则使用 192.0.2.0/24 地址空间。依据该图，如果我的笔记本计算机的 IP 地址是 10.1.0.1，则我可通过内部 IP 网络与使用 IP 地址 10.2.0.2 的我的同事进行通信。当我访问因特网时，为了将我的私有地址 10.1.0.1 映射到一个公开地址（例如

⊖ 从技术角度来说，采用因特网上的虚拟专用网（VPN）或隧道，带有私有地址的流量可穿越因特网，但在两端接入因特网的隧道端点却需要使用公开 IP 地址。

192.0.2.108），则我的报文需要通过防火墙/NAT设备进行路由。在NAT设备中维护映射状态，它修改IP首部，执行的操作是：针对送出报文使用192.0.2.108替换10.1.0.1，针对进入报文，则执行反向替换。

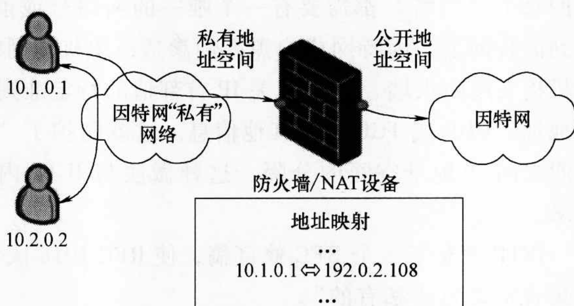


图 1-6 NAT的使用范例：将私有地址映射到公开地址

从寻址容量需求角度来看，我所在机构仅需要足够的IP地址来支持这些特定的互联网到因特网连接以及因特网可达主机（例如Web服务器或电子邮件服务器）。相比于每台内部路由器和外部路由器、服务器和主机都要配置地址的IP地址空间需求而言，这个数量通常要小得多。因为企业要求的是远小得多的公开地址空间，所以私有地址空间的实施，就极大地降低了对地址空间容量的压力。

1.3 无类别寻址

延长IPv4寿命且投入实施的第二项策略是CIDR的实施，它极大地提高了网络分配的效率。可变长度子网掩码法允许在非字节边界对一个有类别网络实施子网划分，和这种做法类似，CIDR支持基础地址块（由一个区域因特网注册机构（RIR）或因特网服务提供商（ISP）分配的）的网络前缀变化的情况。因此，如四个C类（/24）组成的一个连续组可被组合为单一（/22）网络，分配给一个因特网服务提供商。图1-7对此进行了说明。如果如图1-7所示的四个连续块172.16.168.0/24～172.16.171.0/24可用于分配的话，那么它们可作为单一（/22）网络进行分配，即172.16.168.0/22。

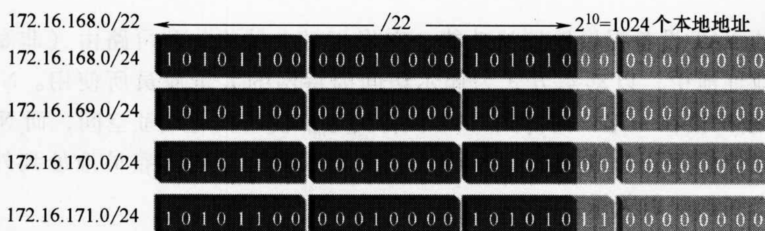


图 1-7 CIDR分配范例

注意较黑阴影比特表示网络号，即前22bit，在所有四个组成网络上是相同的。剩下的10bit表示可用于主机分配的本地地址空间。因为网络地址是以本地地址字段为全0表示的，所以/22网络是以起始的比特串标记的，即172.16.168.0/22。如您所看到的，在非字节边界上计算网络地址所要求的十进制到二进制算术运算方面，CIDR是非常类似于VLSM的。在非字节边界掩码之外，针对本地地址填充0的附加

步骤，有可能引入错误。另外，可将 VLSM 应用到 CIDR 分配的做法，又进一步增加了出现错误的机会。但正如通常的情况一样，要得到更多的灵活性，就要付出代价。CIDR 和 VLSM 扫除了网络类别造成的障碍，从而提供了真正灵活的网络分配和子网划分。

1.4　特殊用途地址

除了私有空间外，针对特殊目的或归档（documentation）的需求，已经预留了某些部分的 IPv4 地址空间。这种 IPv4 地址分配包括预留特殊用途的 IP 地址，这在下面汇总给出，是在 RFC 3330[8] 定义的，在 RFC 5735[9] 中作了更新（修改）。

地址空间	特 殊 用 途
0.0.0.0/8	“这个(本)”网络;0.0.0.0/32 表示在这个(本)网络上的这台主机
10.0.0.0/8	私有 IP 地址空间,依据 RFC 1918,该空间中的地址在公开因特网上是不可路由的
127.0.0.0/8	被指派用作因特网主机回环地址,即 127.0.0.1/32
169.254.0.0/16	“链路本地”地址块,用于 IPv4 自动配置,该地址的目的是在单条链路上进行通信
172.16.0.0/12	私有 IP 地址空间,依据 RFC 1918,该空间中的地址在公开因特网上是不可路由的
192.0.0.0/24	保留,用于 IETF 协议指派
192.0.2.0/24	指派为“Test-Net-1”,用于文档和样例代码
192.88.99.0/24	分配用作 6to4 中继任意播地址(第 17 章对此进行了详细讨论)
192.168.0.0/16	私有 IP 地址空间,依据 RFC 1918,该空间中的地址在公开因特网上是不可路由的
198.18.0.0/15	分配用于网络互连设备的基准测试
198.51.100.0/24	指派为“Test-Net-2”,用于文档和样例代码
203.0.113.0/24	指派为“Test-Net-3”,用于文档和样例代码
224.0.0.0/4	分配用于 IPv4 组播地址指派(以前的 D 类地址空间)
240.0.0.0/4	保留,用于未来用途(以前的 E 类地址空间)
255.255.255.255/32	在一条链路上的受限广播

第2章 IPv6（因特网协议版本6）

2.1 引言

在20世纪90年代早期，因特网被疯狂地用作最流行的全球通信工具，使世界范围内的组织机构都涌向因特网注册机构，请求IP地址空间。IP地址空间需求上的这次冲击波，促使仿照了IETF，因特网工程和标准组织定义了新版本的因特网协议，该新版本协议将提供更多的寻址容量，可满足当时以及可预期未来的地址需求。如第1章所讨论的，诸如CIDR和私有地址空间等技术的采用，有助于消除公开地址空间请求的洪流；但是，可预料这些措施仅延长了IPv4地址空间可使用的时间，即延长10年左右。

IPv4地址空间的可用量持续减少（即剩下的地址越来越少），每个区域因特网注册机构（RIR）不厌其烦地向因特网共同体发出通告，即可使用的IPv4空间是有限的，并将在“数年”内被用光。RIR负责向因特网服务提供商分配IP地址，接下来因特网服务提供商向企业、服务提供商和任何需要IP地址空间的组织机构分配空间。最终，地址耗尽的后果将影响需要公开IP地址空间的机构。各位看到，微软的Vista™、Win7和Server2008等产品都默认地支持IPv6。随着采用Vista或Win7，IPv6也许比人们想象得要更早到来，不管你喜欢与否，它都将到来！



图2-1 首部和报文概念中的IP通用部分

IPv6（因特网协议版本6㊀）是从IPv4（因特网协议版本4）演化发展而来的，但本质上是与IPv4不兼容的。第15章描述了（IPv6与IPv4的）几项迁移和共存技术。IPv6的主要目标是，基于IPv4过去20年的经验，从根本上重新设计IPv4。在过去数年中，添加到IPv4协议族的真实应用支持能力，从IPv6设计一开始就进行了考虑。这包括对安全、组播、移动性和自动配置的支持。

从IPv4演化发展到IPv6过程中，最引人注目的差异是对IP地址字段长度的极大扩展。IPv4使用一个32bit的IP地址字段，IPv6则使用一个128bit的IP地址字段。一个32bit地址字段提供最多2^{32}个地址或42亿个地址。一个128bit地址字段提供2^{128}个地址或340万亿万亿万亿（340×10^{36}）个地址或340undecillion㊁（3.4×10^{38}）个地

㊀ IPv5（IP版本5）从来就没有作为IP的一个官方版本加以实施。在IP首部中的版本号“5”被指派为表示携带所谓ST（因特网流协议）的一个试验实时流协议的报文。如果要更多了解ST，请参见RFC 1819（169）。

㊁ 我们使用美国方式对undecillion的定义即10^{36}，而不是英国方式的10^{66}。

址。为了给这个极其庞大的数字提供形象描述，考虑如下这个数量的 IP 地址。

1）假定地球上有 65 亿个人，则每个人平均有 5×10^{28} 个 IP 地址。

2）地球表面平均每平方英寸有 4.3×10^{20} 个 IP 地址。

3）到达距离我们 250 万光年的最近星系——仙女座的距离上平均每纳米有约 1400 万个 IP 地址。

就像 IPv4 一样，由于子网分配的低效，并不是每个地址都一定会是有用的，但考虑浪费数个地址也不会具有太大影响。除了 IP 地址的这个似乎无限数量外，在 IPv6 和 IPv4 之间存在许多相似点。例如，从基本层面看，就报文首部和内容的概念（见图 2-1）而言，和 IPv4 一样，“IP 报文”这个概念也同样适用于 IPv6，其他的基本概念如协议分层、报文路由以及 CIDR 分配等也是如此。在本章中，我们将焦点放在所定义的 IPv6 地址的种类方面，在下一章讨论 IPv6 子网划分和分配技术。

2.1.1　IPv6 关键功能特征

作为 IPv4 的演化发展结果，IETF 尝试开发 IPv6。从 IPv4 到 IPv6 迁移过程中的演化发展策略的目的是，使 IPv6 能够提供许多新的功能特征，而同时构建于使 IPv4 如此成功的基础概念之上。IPv6 的主要功能特征包括如下内容。

1）扩展的寻址方式。为了改善扩展性，128bit 层次化地指派带有地址范围（本地链路范围以及全局范围）。

2）路由。严格层次化的路由，支持路由汇聚。

3）性能。简单的（不可靠的、无连接的）数据报文服务。

4）扩展能力。新的灵活的扩展首部，为新的首部类型和更加高效的路由提供了固有的扩展能力。

5）多媒体。流标签首部字段有利于实现服务质量（QoS）支持。

6）组播。替换广播，并是必选的。

7）安全。固有的认证鉴权和加密。

8）自动配置。IP 设备可执行无状态地址自配置和有状态地址自配置。

9）移动能力。提供移动 IPv6 支持。

2.1.2　IPv6 首部

IPv6 首部结构布局如图 2-2 所示。虽然源 IP 地址字段和目的地 IP 地址字段的长度都四倍于 IPv4 地址字段长度，但总的 IP 首部长度仅是 IPv4 首部长度的两倍。IPv6 首部中的各字段如图 2-2 所示。

1）版本（Version）。因特网协议版本，在这种情况下是 6。

2）流量类别（Traffic Class）。这个字段替换了 IPv4 的服务类型/DS 首部字段，为请求路由处理而指明流量的类型或优先级。

3）流标签（Flow Label）。标识这个报文所属的一个源和目的地之间流量“流”（由源设定）。这样做的意图是，支持一个给定通信会话内部（例如在一个实时传输以及一个尽力而为数据传输内部的那些会话）针对报文的高效和一致的路由处理。

4）净荷长度（Payload Length）。指明IPv6净荷的长度，即在基本IPv6首部之后的报文部分，是以字节为单位表示的。如果包括扩展首部的话，那么扩展首部被看做净荷的组成部分，并被计算在这个长度参数之内。

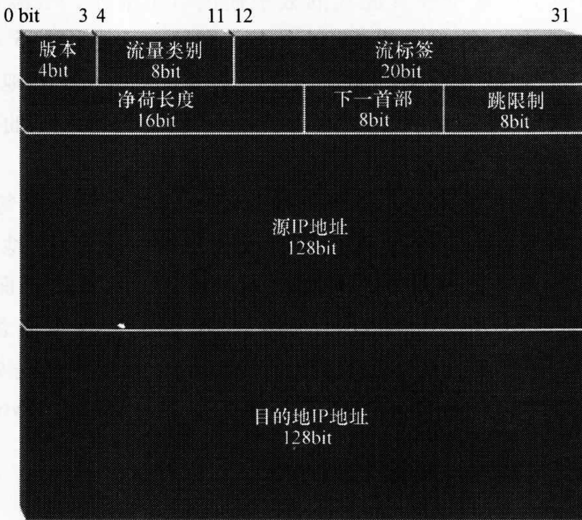


图2-2 IPv6首部[10]

5）下一首部（Next Header）。这个字段指明这个IP首部后跟的首部类型。可能是一个高层协议首部（例如TCP、ICMPv6（因特网控制报文协议版本6）等）或一个扩展首部。仅当源路由、分段、选项以及与该报文相关的其他参数为必要情况下，扩展首部概念才明确指定，而不是像在IPv4中一样作为所有报文上的额外负担。

6）跳限制（Hop Limit）。类似于IPv4 TTL（生存时间）字段，这个字段指定在该报文被丢弃之前可能穿越的跳数。在转发该报文时，每个路由器将这个首部字段的数值减1。

7）源IP地址（Source IP Address）。这条报文发送者的IPv6地址。

8）目的地IP地址（Destination IP Address）。这条报文预期接收者（可能是多个接收者）的IPv6地址。

2.1.3 IPv6寻址㊀

定义了三种类型的IPv6地址。和IPv4中的情况一样，这些地址是应用到接口的，而不是应用到节点的。因此，带有两个接口的一台打印机将可由它的任一接口寻址。可通过任一接口到达该打印机，但该打印机节点本身㊁并没有一个IP地址。当然，对于尝试访问一个节点的终端用户而言，DNS通过使一个主机名（hostname）映射到一个或多个接口地址，就能够隐藏这个微妙差异。

1）单播　单一接口的IP地址。这类似于一个IPv4主机地址的常见解释（非组播/非广播的/32 IPv4地址）。

2）任意播。用于一组接口（通常属于不同节点）的一个IP地址，任何一个接口都是报文的预期接收者。目的地为一个任意播地址的一条报文，被路由到配置了该

㊀ 本章的介绍性章节主要参考了文献［11］第2章的资料。

㊁ 许多路由器和服务器产品，通过一个软件回环地址，支持一个“设备（Box）地址”的概念。不要与127.0.0.1或::1回环地址相混淆，这个回环地址支持到任一设备接口的可达性。

组播地址的最近距离的接口（依据路由表度量指标衡量）。这里的理念是，发送者不必关心那台特定主机或接口接收该报文，但是确实是共享该任意播地址的那些主机或接口接收了这条报文。任意播地址是从单播地址分配的相同地址空间中分配的。因此，人们不能从表面上在一个单播地址和一个任意播地址之间做出区分。在提供类似的到预期所使用服务（例如针对 DNS 服务器，使用一个共享的单播 IPv4 地址）的最近路由方面，任意播在 IPv4 网络中最近产生了争议。在简化客户端配置（不管客户端连接到所在网络的哪一部分），使客户端总是使用相同［任意播］IP 地址查询一台 DNS 服务器方面，这种做法提供了便利。在第 11 章中我们将讨论使用任意播地址的 DNS 部署。

3）组播。一组接口（典型情况下属于不同节点）的一个 IP 地址，所有节点都是报文的预期接收者。这当然类似于 IPv4 组播。不像 IPv4 的是，IPv6 不支持广播。相对而言，在 IPv4 中使用广播的应用，例如 DHCP，在 IPv6 中使用组播到一个周知（即预定义的）DHCP 组播组地址的方法。

一个设备接口可能具有任何地址类型或所有地址类型的多个 IP 地址。IPv6 也定义了 IP 地址的一个链路本地范围，来唯一地标识连接到一条特定链路（例如一个 LAN（局域网））的各个接口。例如，可在每个站点或每个机构范围内，从管理角度定义附加的地址范围，我们将在本章后面讨论这一点。

2.1.4　地址表示法

回顾一下，IPv4 地址是以点分十进制格式表示的，其中 32bit 的地址被分成四个 8bit 段，每个段转换成十进制数，之后以“点”分隔。如果你认为记忆四个十进制数的一个串是困难的，那么 IPv6 将会使你有点苦不堪言。IPv6 地址不是以点分十进制表示法表述的，它们是使用一个冒号分隔的十六进制格式加以表示的。首先从比特级别开始分，128bit IPv6 地址被分成八个 16bit 段，每个段被转换为十六进制，之后以冒号分隔。每个十六进制“数字”表示 4bit，转换规则依据为，每个十六进制数字（0～F）到其 4bit 二进制数值的映射如下所示。每个十六进制对应于具有如下可能数值的 4bit。

0 = 0000	4 = 0100	8 = 1000	C = 1100
1 = 0001	5 = 0101	9 = 1001	D = 1101
2 = 0010	6 = 0110	A = 1010	E = 1110
3 = 0011	7 = 0111	B = 1011	F = 1111

在将一个 128bit IPv6 地址从二进制转换为十六进制之后，我们将每四个十六进制数字归为一组，并以冒号将它们分隔开。我们使用名词“尼伯”（nibble）[⊖]来代表四个十六进制数字或 16bit 的分组；因此，我们得到八个由冒号分隔的尼伯数值，产生看起来如图 2-3 所示的一个 IPv6 地址。

在 IPv4 中，要处理四个十进制数值，相互之间以点分隔，每个数值在 0～255 之

⊖ 尼伯（nibble），本书中和常用的含义（通常指 4 个二进制数）不同。——译者注

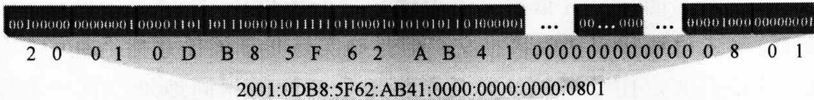


图 2-3 IPv6 地址：二进制转换为十六进制[11]

间，与此不同的是，IPv6 地址由多达八个十六进制数值组成，由冒号分隔，每个数值在 0 ~ FFFF 之间。当书写 IPv6 地址时，有两种可接受的缩写形式。第一种形式是，在一个尼伯分段内部，即冒号之间的前导零可被去掉。因此，上述地址可缩写为：

2001：DB8：5F62：AB41：0：0：0：801

缩写的第二种形式是，使用双冒号表示一个或多个连续的零尼伯组。使用这种缩写形式，上述地址可进一步缩写为：

2001：DB8：5F62：AB41 ：: 801

这难道不好得多了吗？注意在一个地址表示内部仅可使用一个双冒号。因为在地址中总是存在八个尼伯分段，对于一个双冒号表示法，人们可容易地计算它们中有多少个为零；但是，对于一个以上的双冒号，就将存在二义性问题。

考虑地址 2001：DB8：0：56FA：0：0：0：B5。我们可将这个地址缩写为：

2001：DB8 ：: 56FA：0：0：0：B5 或 2001：DB8：0：56FA ：: B5

我们可容易地计算出，在第一种情形中，双冒号表示一个尼伯（总的 8 个尼伯减去如图 2-3 所示的 7 个尼伯），在第二种表示法中，双冒号表示 3 个尼伯（总的 8 个尼伯减去如图 2-3 所示的 5 个尼伯）。如果我们尝试将这个地址缩写为 2001：DB8：：56FA：：B5，则我们不能无二义性地对此解码，因为它可能表示如下地址中的任何一个地址：

2001:DB8:0:56FA:0:0:0:B5

2001:DB8:0:0:56FA:0:0:B5

2001:DB8:0:0:0:56FA:0:B5

因此，要求缩写规则总是成立的，即在一个 IPv6 地址中仅可出现一个双冒号。

2.1.5 地址结构

如图 2-4 所示，将 IPv6 地址分成三个字段。

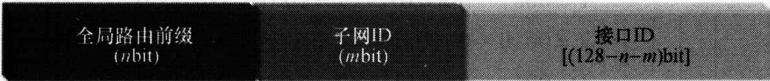


图 2-4 IPv6 地址结构[12]

全局路由前缀与一个 IPv4 网络号近似，由路由器用来将报文转发到本地服务于对应该前缀的网络的路由器（可能是多台路由器）。例如，一个 ISP 的一个客户被指派一个/48 长度的全局路由前缀，则目的地为这个客户的所有报文将包含对应的全局路由前缀值。在这种情形中，依据图 2-4，$n=48$。当表示一个网络时，写出全局路由前缀，后跟斜杠，之后是网络规模大小，称为前缀长度。假定我们的一个范例 IPv6 地址 2001：DB8：5F62：AB41：：801，存在于一个/48 的全局路由前缀内部，则这个

前缀地址将被表示为 2001：DB8：5F62：：/48。和 IPv4 的情形一样，除前缀长度外带有零值比特（在这种情形下，是 bit49～128）的网络地址被表示为终结的双冒号。

子网 ID 提供了在组织机构内部表示特定子网的一种方式。我们的拥有一个/48 前缀长度的 ISP 客户，选择使用 16bit 表示子网 ID，这样就提供了 2^{16} 或 65534 个子网。在这种情形中，依据图 2-4，$m=16$。这样下来，留给接口 ID 的就是（128-48-16）bit = 64bit。接口 ID 表示报文的源或预期接收者的接口地址。这就和我们后面将讨论的一样，迄今为止，已分配使用的全局单播地址空间要求一个 64bit 的接口 ID 字段。

在将一个网络 ID（由全局路由前缀和子网 ID 组成）与一个接口 ID 分隔（区分开）过程中，这种 IPv6 地址结构独特之处之一是，一个设备可保留相同的接口 ID，而不用管它连接到了哪个网络，有效地将你的接口 ID（“你是谁”）和你的网络前缀（“你在哪里”）区分开来。正如我们将看到的，这种做法有利于实现地址自动配置，虽然它没有考虑隐私问题。但我们稍稍有点超前了（即过早地讲了一些内容），所以让我们跳转回到宏观层次，并考虑迄今为止由因特网地址管理权威（因特网编号管理局（IANA））所分配的 IPv6 地址空间方面。

2.2 IPv6 地址分配

在表 2-1 中以暗灰色突出显示迄今为止由 IANA 分配的地址空间，并在接下来的文字中加以讨论。这些地址分配代表的地址空间要略小于总可用 IPv6 地址空间的 14%。

表 2-1 IPv6 地址分配[13]

IPv6 前缀	二进制形式	IPv6 空间的相对尺寸	分配
0000::/3	000	1/8	由 IETF 保留：“未指派地址”(::)和回环地址(::1)就是从这个地址块中指派的
2000::/3	**001**	**1/8**	**全局单播地址空间**
4000::/3	010	1/8	由 IETF 保留
6000::/3	011	1/8	由 IETF 保留
8000::/3	100	1/8	由 IETF 保留
A000::/3	101	1/8	由 IETF 保留
C000::/3	110	1/8	由 IETF 保留
E000::/4	1110	1/16	由 IETF 保留
F000::/5	1111 0	1/32	由 IETF 保留
F800::/6	1111 10	1/64	由 IETF 保留
FC00::/7	**1111 110**	**1/128**	**唯一本地单播**
FE00::/9	1111 1110 0	1/512	由 IETF 保留
FE80::/10	**11111 1110 10**	**1/1024**	**链路本地单播**
FEC0::/10	1111 1110 11	1/1024	由 IETF 保留
FF00::/8	**1111 1111**	**1/256**	**组播**

2.2.1　::/3——保留地址

前缀为 $[000]_2$ 的地址空间目前由 IETF 保留。在这个地址空间内部且具有独特含义的地址包括非指定（::）地址和回环（::1）地址。IPv6 寻址架构规范 RFC 4291[12]要求，除了在这个地址空间内部的那些地址（以::/3（$[000]_2$ 开始））外，所有单播 IPv6 地址都必须使用一个 64bit 的接口 ID 字段，且这个接口 ID 字段必须利用修订的 EUI-64㊀算法，将接口的层 2 地址或硬件地址映射到一个接口 ID。因此，在::/3 地址空间内部的地址，可具有任意长度的接口 ID 字段，这点不像除此之外其他部分的 IPv6 单播地址空间，在其他部分的地址空间中必须利用一个 64bit 的接口 ID 字段。

2.2.2　2000::/3——全局单播地址空间

迄今为止被分配的全局单播地址空间 2000::/3 表示了 2^{125} 或 4.25×10^{37} 个 IP 地址。考虑到在 IPv6 地址结构［RFC 4291[12]］中定义的 64bit 接口 ID 要求，全局单播地址格式形式化地定义为 RFC 3587[14]，如图 2-5 所示。

前 3bit 是 $[001]_2$，指明是全局单播地址空间。接下来的 45bit 构成全局路由前缀，接着分别是 16bit 子网 ID 和 64bit 的接口 ID。当前的指导准则呼吁各 ISP 将/48 型网络分配给他们的客户，如此就将全局路由前缀分配到了客户手中。那么每个客户，通过在剩下的 16bit 子网 ID 字段内部为每个子网唯一分配指派值的方法，可定义多达 65534 个子网。

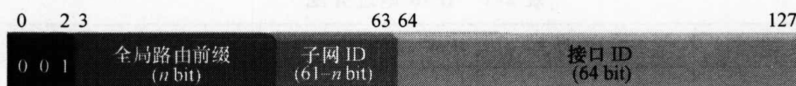


图 2-5　全局单播地址格式[14]

2.2.3　FC00::/7——唯一本地地址空间

在 RFC 4193[15]中定义的本地唯一地址（ULA）空间，其目的是通常在一个站点内部提供本地可分配和可路由的 IP 地址。RFC 4193 陈述说“期望这些地址在全球因特网上是不可路由的”。因此虽然不像 RFC 1918 在定义私有 IPv4 地址空间时那么严格，但本地唯一地址空间本质上仍然是私有地址空间，提供“本地”（局部）寻址，它以较高概率保持仍然是全球唯一的。本地唯一地址空间的格式如图 2-6 所示。

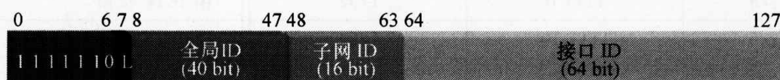


图 2-6　本地唯一地址空间的格式[15]

㊀ EUI-64 指由 IEEE 定义的 64bit 扩展独特标识符。我们将在本章稍后部分讲解修订的 EUI-64 算法。

前7bit，即比特0~6，是 $[1111\ 110]_2$ = FC00::/7，识别该地址是一个本地唯一地址。第8个bit，即“L”bit，如果全局ID是本地指派分配的，则设置为“1”；将“L”bit设置为“0”的含义当前还未确定，但因特网任务工程组（IETF）已经讨论过，拟将这个设置用于全局本地唯一地址，它是通过因特网地址注册机构分配的。40bit长的全局ID字段的目的是表示一个全局唯一前缀，并必须使用一个伪随机算法（不是顺序方法）进行分配。无论在哪种情形中，得到的/48前缀构成了组织机构的ULA地址空间，从该空间中可分配用于内部使用的子网。子网ID是一个16bit的字段，用来识别每个子网，而接口ID是一个64bit字段。

在RFC 4193中描述了得到一个全局唯一ID的一种范例性的伪随机方法，它建议以如下方式计算一个散列数值[⊖]。

1）一台网络时间协议（NTP）服务器以64bit NTP格式报告的当前时间。

2）与实施该算法主机上一个接口的一个EUI-64接口ID拼接在一起。

之后将散列运算结果的最低（最右边的）40bit构成全局ID。

2.2.4 FE80::/10——链路本地地址空间

链路本地地址仅用在一条特定链路上，例如一条以太网链路；带有链路本地目的地址的报文是不可路由的。即，具有链路本地地址的报文将不能到达对应链路外面（即范围为对应链路）。这些地址用于地址自动配置和邻居发现，这将在后面讨论。链路本地地址空间的格式如图2-7所示。

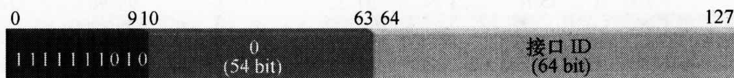


图2-7 链路本地地址空间的格式[12]

FE80::/10链路本地前缀后跟54个零bit和64bit的接口ID。

2.2.5 FF00::/8——组播地址空间

组播地址识别典型情况下在不同节点上的一组接口。可将组播地址想象为一个范围受限的广播。所有组播组成员都使用相同的组ID，因此所有成员将接受目的地为组播组的报文。一个接口可能有多个组播地址；即，它可能属于多个组播组。IPv6组播地址空间的格式如图2-8所示。



图2-8 组播地址空间的格式[12]

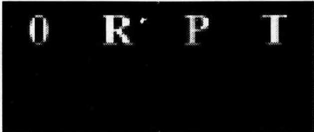
前缀FF00::/8标识一个组播地址。下一个字段是一个称为“标志”（flags）的

⊖ 通过在要被散列的数据和一个随机数值上执行一个数学运算，得到一个散列数值。在这种情形中，要求使用一个特定的数学算法，即安全散列算法1或SHA-1。

4bit 字段。组播地址的格式取决于标志字段的数值。范围（scope）（也许被带有感情色彩地称作“scop”（吟游诗人））字段表明组播范围的广度，无论是每节点、链路、全局还是其他范围数值，都将在下面定义。幸运的是，标志和范围字段的数值均可容易地通过分别查看地址内部的第 3 个十六进制和第 4 个十六进制数字做出区分，我们将在稍后总结一下。

1. 标志（flags）

标志字段由 4bit 组成，我们将从右到左加以讨论（12）。



（1）*T*bit 指明组播地址本质上是临时用途的，还是由 IANA 分配的一个周知地址。*T*bit 定义如下。

1）$T=0$。这是一个 IANA 分配的周知组播地址（见图 2-9）。在这种情形中，这个 112bit 的组播地址是一个 112bit 的组 ID 字段。



图 2-9　带有标志 $T=0$ 的组播地址

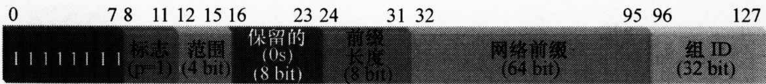


图 2-10　带有标志 $P=1$ 的组播地址[16]

迄今为止，IANA 已经分配了许多组 ID㊀。例如，组 ID = 1 指代在关联范围（由范围字段定义）内部的所有节点，组 ID = 2 指代在范围内的所有路由器等。范围字段是如下定义的，周知的组播地址范例如下。

① F01：：1 = 本链路上的所有节点。

② FF02：：2 = 本链路上的所有路由器。

③ FF05：：1 = 本站点上的所有节点。

④ FF05：：2 = 本站点上的所有路由器。

2）$T=1$。这是一个临时分配的或短暂的组播地址。它可以是为一个特定组播会话或应用分配的地址。一个范例如 FF12：：3：F：10。

（2）*P*bit 指明组播地址是否部分地由一个对应的组播地址前缀组成。*P*bit 定义如下㊁。

1）$P=0$。这个组播地址不是依据网络前缀进行分配的。带有 $P=0$ 的一条组

㊀ 请参见 http：//www. iana. org/assignments/ipv6-multicast-addresses，了解最新分配情况。

㊁ 在 RFC 3306[16] 中可见到 *P*bit 定义的文字描述。

播报文的格式见上面的描述（即当 $T=0$ 时），带有 112bit 的组 ID 字段。

2）$P=1$。这个组播地址是如下分配的：依据“拥有”组播地址分配能力的单播子网地址的网络前缀进行分配。为了进行比较简单的管理，这使与所分配单播空间关联的组播空间分配成为可能。如果 $P=1$，则 Tbit 也设置为 1。一条组播报文的对应格式如图 2-10 所示。

当 $P=1$ 时，范围字段后跟 8 个零 bit（保留）、一个 8bit 前缀长度字段以及一个 64bit 网络前缀字段和一个 32bit 组 ID 字段。前缀长度字段代表所关联单播网络地址的前缀长度。网络前缀字段包含对应的单播网络前缀，而组 ID 字段包含关联组播组 ID。

例如，如果一个单播地址 2001：DB8：B7：：/48 被分配给一个子网，一个对应的基于单播的组播地址将具有这样的形式，即 FF3s：0030：2001：DB8：B7：：g，其中

1）FF = 组播前缀。

2）$3=[0011]_2$，即 $P=1$、$T=1$。

3）s = 一个有效的范围，我们将在下一节定义。

4）00 = 保留比特。

5）在我们所举例子中的前缀长度，30 = 十六进制表示的前缀长度 = $[0011\ 0000]_2$ = 十进制表示的 48。

6）2001：DB8：B7：0 = 2001：0DB8：00B7：0000 = 64bit 网络前缀字段中的 48bit 网络前缀。

7）g = 一个 32bit 的组 ID。

当前缀长度字段 = FF 且 $s \leqslant 2$ 时，这种格式出现一种 $P=T=1$ 的特殊情形。在这种情形中，网络前缀字段不是由单播网络地址组成的，而是将由相应接口的接口 ID 组成的。为了确保接口 ID 的唯一性，所用接口 ID 必须已经通过重复地址检测（DAD）过程，在本章后面讨论 DAD 过程。在这种特殊情形中，范围字段必须为 0、1 或 2，这意味着接口本地范围或链路本地范围。在 RFC 4489[17] 中，将这种链路范围的组播地址格式定义为 IPv6 地址结构的一个扩展。

（3）标志字段中的 Rbit 支持一个组播会聚点（RP）的指定，RP 支持将成为组播组署名用户（成员），在永久地加入到组之前，先临时地连接进来。如果 Rbit 被设置为 1，那么 P 和 Tbit 也必须设置为 1。当 $R=1$ 时，组播地址是基于一个单播前缀的，但 RP 接口 ID 也要加以指定（见图 2-11）。$R=1$ 时的组播地址格式等同于 $R=0$ 和 $P=1$ 时的情形，例外是保留字段被分割成一个 4bit 保留字段和一个 4bit 会聚点接口 ID（RIID）字段。

1）RP 的 IP 地址是通过将相应前缀长度的网络前缀与 RIID 字段的数值串接得到的。例如，如果在该［单播］网络上的一个 RP 是 2001：DB8：B7：：6，则关联的组播地址将是 FF7s：0630：2001：DB8：B7：g，其中 s = 下面定义的一个有效范围，g = 一个 32bit 的组 ID。

2）这个地址的显式分解说明如下。

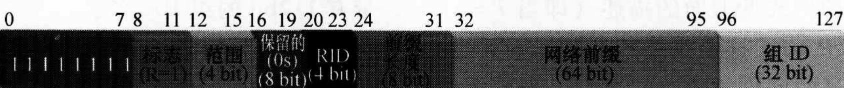


图 2-11 带有标志 $R=1$ 的组播地址

① FF = 组播前缀。

② 7 = $[0111]_2$，即 $R=1$、$P=1$ 和 $T=1$。

③ s = 下面定义的一个有效范围。

④ 0 = 保留比特。

⑤ 6 = RIID 字段，将被附加在网络前缀字段后面。

⑥ 在我们所举例子中的前缀长度，30 = 以十六进制表示的前缀长度 = $[0011\ 0000]_2$ = 十进制的48。

⑦ 2001：DB8：B7：0 = 2001：0DB8：00B7：0000 = 64bit 网络前缀字段中的48bit 网络前缀。

⑧ g = 一个 32bit 的组 ID。

3）第一个标志比特保留，并被设置为0。

2. 组播标志小结

谁能想到组播寻址可能会这样复杂呢？但这就和典型情况一样，随复杂性而来的是灵活性！总结一下，上述比特规定的净结果产生了当前定义标志字段的如下有效数值。因为标志字段直接跟在开始的 8 个 “1” bit 之后，所以我们将 “有效前缀” 表示为开始的 8bit 后跟有效的 4bit 标志字段（见表 2-2）。

表 2-2 组播标志小结

标志(二进制)	有效的前缀	解　释
0000	FF00::/12	永久地分配 112bit 组 ID，其范围受到 4bit 范围字段的约束
0001	FF10::/12	临时地分配 112bit 组 ID，其范围受到 4bit 范围字段的约束
0011	FF30::/12	临时地分配基于单播前缀的组播地址
0111	FF70::/12	临时地分配基于单播前缀的组播地址，带有会聚点接口 ID
所有其他标志数值	—	未定义

3. 范围（Scope）

范围字段确定组播地址的范围或“所及范围”（自然是足够大的）。这由路由器使用，用相应的范围沿组播路径约束组播通信的所及范围。注意，为了增强相应所及范围的约束，除了接口本地、链路本地和全局外的范围必须采用管理方式，在服务给定范围的路由器上加以定义。表 2-3 简单总结了有效的范围数值。

2.2.6 特殊情形的组播地址

（1）被请求的（solicited）节点组播地址。每个节点必须支持的一种组播地址形式是被请求的节点组播地址。这个地址用于地址自动配置的重复地址检测阶段过程和

邻居发现协议，该协议使一条链路上识别 IPv6 节点成为可能。通过将被请求节点的接口 ID 最低（最右边）24bit 添加到周知的 FF02::1:FF00/104 前缀之后，形成被请求的节点组播地址。

表 2-3　组播范围字段解释

范围字段			
二进制	十六进制	含义(范围)	描　述
0000	0	保留	保留
0001	1	接口本地	由一个节点上单接口组成的范围,仅用于回环传输
0010	2	链路本地	组播报文在其上传输的链路范围
0011	3	保留	保留
0100	4	管理本地	受限在管理上配置的最小范围。这个范围没有依据物理连接性或其他组播有关的配置
0101	5	站点本地	范围受限于管理上确定的站点
0110、0111	6、7	未指派	未用
1000	8	组织机构本地	依据管理上确定的一个组织机构内部的多个站点组成的范围
1001 ~ 1101	9 ~ D	未指派	未用
1110	E	全局范围	不受限制的范围
1111	F	保留	保留

例如，我们假定一个节点拟解析 IP 地址为 2001:DB8:4E:2A:3001:FA81:95D0:2CD1 的设备（接口）的链路层地址。使用最低 24bitD02CD1（十六进制），该设备可将其请求发送到 FF02::1:FFD0:2CD1（见图 2-12）。

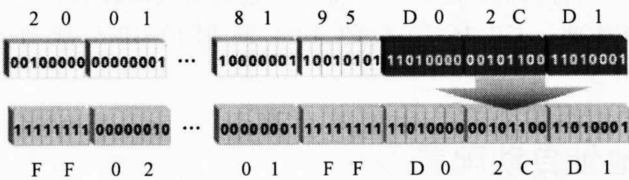


图 2-12　被请求的节点组播地址形成方法[12]

（2）节点信息查询地址。节点信息查询地址是这样一个组播地址，它支持从一台 IPv6 主机请求主机名、IPv6 和 IPv4 地址信息（见图 2-13）。如果您认为这听起来有点与 DNS 已经提供的功能有所重叠，那您就猜对了。但是，依据 RFC 4620[18]，解析的这种模式“目前限于诊断和调试工具及网络管理”。要得到这种信息，该查询并不查询一台 DNS 服务器，而是将一条查询发送到节点信息表查询地址。

这种组播地址格式的使用，支持一个 IPv6 地址仅基于预期接收者的主机名即可形成；如果 IPv6 地址已经知道，且请求了主机名信息，则 IPv6 地址本身就可用作目的地址。当针对一个已知的主机名请求 IP 地址信息时，使用 128bit MD-5 算法对规范

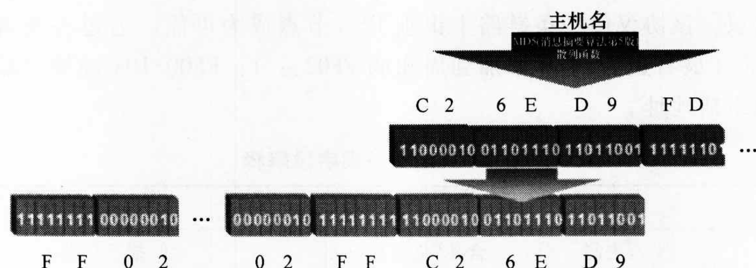


图 2-13 被请求的节点信息查询地址

的主机名⊖进行散列运算，将从散列运算得到的前 24bit 添加到 FF02::2:FF00:0/104 前缀后面。当每个节点接收到以这个节点信息查询地址为目的地的消息时，它将地址中的最后 24bit 与其自身主机名计算出的散列数值的前 24bit 进行比较；如果两者匹配的话，则接收者将以被请求的信息做出应答。

2.2.7 带有内嵌 IPv4 地址的 IPv6 地址

在第 15 章中，我们将讨论 IPv4 到 IPv6 的迁移和共存策略，但下面我们将先介绍一下 IPv4 映射的 IPv6 地址（见图 2-14）。这种类型的地址在因特网上是不可路由的，仅可由一些（地址）转换方案使用，且一般情况下，不应该用在一条通信链路的 IPv6 报文内部。这个地址格式由 80 个 0bit，后跟 16 个 1bit，再后跟 32bit 的 IPv4 地址组成。

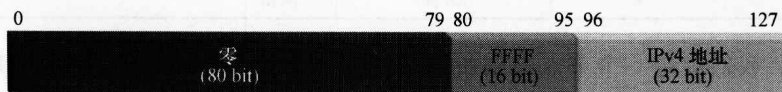


图 2-14 IPv4 映射的 IPv6 地址[12]

这种地址表示法将人们熟悉的 IPv4 点分十进制格式添加到指定的 IPv6 前缀之后进行了组合。因此，172.16.20.5 的 IPv4 映射的 IPv6 地址将被表示为::FFFF:172.16.20.5。

2.3 IPv6 地址自动配置

IPv6 被宣称的优势之一是设备可自动配置它们自身 IPv6 地址的能力，对于设备当前正在连接的子网⊖而言，该地址是独特的，且与该子网有关。有三种基本形式的 IPv6 地址自动配置。

1）无状态的。这个过程是“无状态的”，原因是它不依赖于外部分配机制（例

⊖ 从技术角度而言，“规范的主机名”是以小写字母表示的完全符合要求的域名的第一个“标签”。在第 9 章将详细描述这个术语，但如果说这一般是预期发送到的目的地主机名，也是可以的。

⊖ 注意一些 IPv4 协议栈，例如在众多协议栈中微软 Windows 2000 和 XP 提供的协议栈，使用 IPv4 “链路本地”地址空间 169.254.0.0/16 实施地址自动配置。

如 IPv6 动态主机配置协议（DHCPv6））的状态或是否存在。在没有外部或用户干预的情况下，设备尝试配置其自身的 IPv6 地址（可能是多个地址）。

2）有状态的。有状态的过程仅依赖于外部地址分配机制（例如 DHCPv6）。DHCPv6 服务器以类似于 IPv4 DHCP 操作的方式，将 128bit IPv6 地址分配给设备。在第 5 章中将详细描述这个过程。

3）无状态和有状态组合方式。这个过程涉及无状态地址自动配置与其他 IP 参数的有状态配置相结合一起使用的形式。通常情况下，这需要一台设备使用无状态方法自动配置一个 IPv6 地址，之后利用 DHCPv6 得到其他参数或选项，比如要在给定网络上联系哪台 NTP 服务器来查询时间分辨率（resolution）。

在最基本的层次上，一个 IPv6 单播地址的自动配置包括将设备所连接网络的地址（您在哪里）和设备的接口 ID（您是谁）串接在一起的操作。让我们首先考虑设备如何确定它所连接网络的地址。

2.4　邻居发现

在 IPv6 中的邻居发现过程使一个节点能够发现它所连接的 IPv6 子网地址。一般而言，邻居发现也支持在子网上识别其他 IPv6 节点、识别它们的链路层地址、发现服务该子网的路由器（可能是多台路由器）以及实施重复地址检测。路由器的发现使 IPv6 节点能够自动地识别子网上的各路由器，就弱化了在设备的 IP 配置内部人工配置一个默认网关的需求。这个邻居发现功能使一台设备可识别分配给链路的网络前缀（可能有多个前缀）和对应的前缀长度（可能有多个前缀长度）。

发现过程需要每台路由器周期性地在其配置的每个子网上发送通告，该通告指明它的 IP 地址，它提供默认网关功能的能力、它的链路层地址、所服务链路上的网络前缀（可能有多个前缀）（包括对应的前缀长度）和有效的地址寿命，以及其他配置参数。

路由器通告也指明是否存在一台 DHCPv6 服务器可用于地址分配或其他配置。路由器通告中的 *M*bit（被管理地址的配置标志）指明 DHCPv6 服务可用于地址和配置设置。*O*bit（其他配置标志）指明除了 IP 地址之外的配置参数可通过 DHCPv6 得到；这样的信息可能包括对于这条链路上的各个设备可查询哪些 DNS 服务器。各节点也可使用路由器请求消息，请求路由器通告，目的地址要设为链路本地路由器组播地址（FF02：：2）。

2.4.1　改进的 EUI-64 接口标识符

一旦一个节点识别出它所连接到的子网，那么通过形成其接口 ID，它就可完成地址自动配置过程。IPv6 地址结构约定，除了那些以二进制 $[000]_2$ 开始的地址外，所有单播 IPv6 地址必须利用改进的 EUI-64 算法，得到一个 64bit 的接口 ID。“未改进的” EUI-64 算法指将 IEEE 向每个网络接口硬件制造商（例如一个以太网地址的初始 24bit）发行的 24bit 公司标识符与一个 40bit 的扩展标识符串接在一起。对于 48bit 的

以太网地址，以太网地址的公司标识符部分（前 24bit）后跟一个 16bit 的 EUI 标签（定义为 FFFE），再后跟 24bit 的扩展标识符（即以太网地址剩下的 24bit）。

将一个未改进的标识符转换为一个改进的 EUI-64 标识符所需的改进操作，要求逆转公司标识符字段的“*u*”bit（全局/本地比特）。“*u*”bit 是公司标识符字段中从高位数起的第 7 位。因此，一个 48bit MAC 地址的这种算法是，逆转“*u*”bit，并在公司标识符和接口标识符之间插入十六进制数值 FFFE。使用 MAC 地址 AC-62-E8-49-5F-62 的这个过程，如图 2-15 所示，得到的接口 ID 是 AE62：E8FF：FE49：5F62。

对于非以太网 MAC 地址，该算法要求使用链路层地址作为接口 ID，并带有零填充（从“左”开始）。对于没有链路层地址可用的各种情形，例如在一条拨号链路上的情况，建议一个唯一的标识符可利用另一个接口的地址、一个序列号或其他设备相关的标识符。

接口 ID 也许不是唯一的，特别当不是从一个唯一 48bit MAC 地址得到的情况下，尤其可能会出现这种情况。因此，在提交新地址之前，设备必须执行重复地址检测。在完成 DAD 过程之前，认为地址是暂时的。

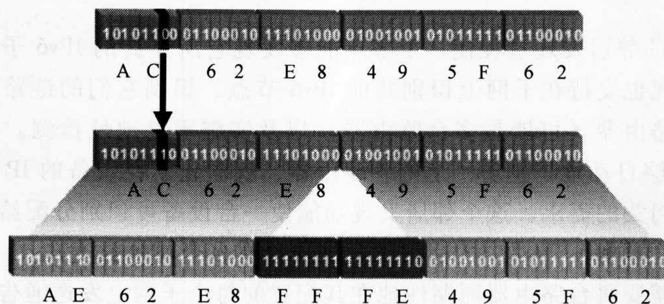


图 2-15　改进的 EUI-64 接口 ID 范例[11]

2.4.2　重复地址检测

使用邻居发现过程实施 DAD，是指为了识别 IP 地址的一个先期占有者，设备向它推算得到的（或从 DHCPv6 得到的）IPv6 地址发送一条 IPv6 邻居请求报文。稍稍延迟之后，设备也向与这个地址关联的被请求节点组播地址发送一条邻居请求报文。

如果另一台设备已经在使用该 IP 地址，它将以一条邻居通告报文做出响应，那么自动配置过程将终止；即要求人为干预或配置该设备使用一个替代的接口 ID。如果没有收到邻居通告报文，那么该设备可假定该地址是唯一的（未被使用），并将其分配给对应的接口。不仅对自动配置的地址，而且对那些静态确定的地址或通过 DH-CPv6 得到的那些地址，都要求参与执行这个过程的邻居请求和邻居通告。

IPv6 地址有一个寿命，在寿命期间，它们是有效的（见图 2-16）。在一些情形中，寿命是无限的，但地址寿命的概念同样适用于 DHCPv6 租赁的地址和自动配置的地址。在便利网络重新编址的过程中，这是有用的。针对每个网络前缀，路由器都被配置一个首选寿命和一个有效寿命数值，在路由器的路由器通告消息中，它们通告为

每个网络前缀通告这两个数值。成功地通过上面所述重复地址检测过程而被证明唯一的 IP 地址，可认为是首选的或过时的（deprecated）。无论是在哪种状态，该地址都是有效的，但这种差异为上层协议（例如 TCP、UDP）提供了选择在后续会话过程中不太可能会发生变化的一个 IP 地址的方式。

依据通告的数值，一台设备采用每条路由器通告中的数值，刷新它的首选时间和有效时间。当一个首选前缀的时间过期时，所关联的地址（可能是多个地址）将成为过时的，虽然仍然是有效的。因此，过时的状态提供了一个过渡时间，在此期间，该地址仍然是可用的，但不应该用之发起新的通信。一旦地址的有效寿命过期，则该地址不再有效，即不能再用。如果一个子网被重新分配一个不同的网络前缀，则路由器可被配置为通告新的前缀，当老的前缀过期时，在网络上的设备将使用新的前缀实施自动配置过程。

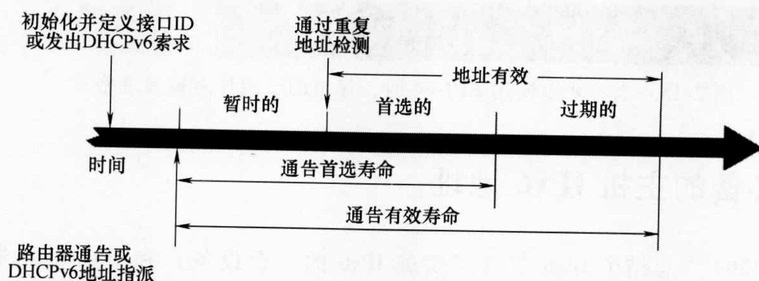


图 2-16 IPv6 地址寿命（该图依据参考文献［19］画出）

2.5 保留的子网任意播地址

RFC 2526[20]为保留的子网任意播地址定义了格式。IPv6 设备使用这些地址将报文路由到一个具体指定子网上一种特定类型的距离最近的设备。例如，一个保留的子网任意播地址可被用来将报文发送到一个具体指定子网上的距离最近的移动 IPv6 家乡代理。因为全局路由前缀和子网 ID 是在这个地址类型内部确定的，所以它使一个节点能够在那个子网上定位期望类型的最近距离的节点。

地址的格式取两种形式之一，依据是子网前缀是否要求以改进的 EUI-64 格式形成接口 ID 字段。回顾一下，所有全局单播地址（除了以［000］$_2$ 开始的那些单播地址外）必须利用 64bit 的接口 ID，是基于接口的链路层地址和前面所描述的 EUI-64 算法形成接口 ID 的。

（1）如果要使用 EUI-64 算法，那么保留的子网任意播地址是通过串接以下字段而形成的（见图 2-17）。

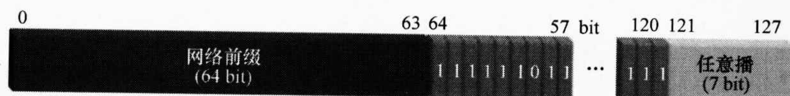


图 2-17 当要求使用 EUI-64 时，保留的子网任意播地址格式[20]

1）64bit 全局路由前缀和子网 ID。

2）除了这个序列中的第 7bit（从左向右数，从 1 开始算起的第 71bit）为 0 外，其他 57bit 都全为 1。当实施 EUI-64 算法时，这个第 7bit 对应于硬件地址中公司标识符字段的“*u*” bit（全局/本地比特）。在这个特定场景中，这个比特总是为 0，表示该比特的“本地”设置。

3）7bit 的任意播 ID。RFC 2526 为移动 IPv6 家乡代理任意播定义单一任意播 ID 为十六进制的 7E。虽然 IANA 可依据未来 IETF RFC 发布版本（publications），分配其他任意播 ID，但目前其他任意播 ID 数值是保留的。

（2）如果 EUI-64 不被要求依据全局路由前缀和子网 ID 生成，那么网络前缀长度中的 *n*bit 就是任意的，后跟 121-*n* bit，再后跟 7bit 任意播 ID（见图 2-18）。

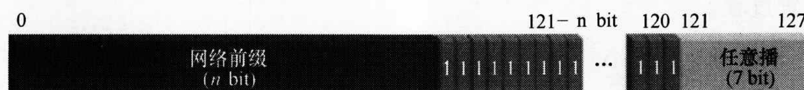


图 2-18　当不要求使用 EUI-64 时，保留的子网任意播地址格式[20]

2.6　必备的主机 IPv6 地址

RFC 4294[21] 总结了 IPv6 节点（实施 IPv6 的一台设备）和 IPv6 路由器的要求。就必备的地址而言，所有 IPv6 节点必须能够自己识别如下 IPv6 地址。

1）回环地址（:: 1）。

2）它的链路本地单播地址（通过自动配置过程配置的 FE80:: <接口 ID>）。

3）所有节点组播地址（FF0*s*:: 1，其中 *s* = 范围）。

4）在每个接口上自动或人工配置的单播和任意播地址。

5）针对它的每个单播和任意播地址得到的被请求节点组播地址。

6）该节点所属每个组播组的组播地址。

要求一台路由器节点支持上述地址，除此之外，还要支持如下地址。

1）子网路由器任意播地址（<子网前缀>:: /128，即接口 ID = 0*s*）。

2）所有路由器组播地址（FF0*s*:: 2，其中 *s* = 范围）。

3）在该路由器上配置的任意播地址。

诸如 DHCP 服务器和 DNS 服务器等其他设备类型，必须识别范围受限的组播地址（对应于 IANA 分配的组 ID（即当标志 = 0 时的情况））。

第3章　IP地址分配

在本章我们将从作为IP地址管理实践基础的技术和应用展开描述。另外，我们将通过范例的方式，展示这些技术和应用。因此以IP地址分配基础知识开始，我们将逐步地将每个新概念应用到一个被称为国际处理和材料（International Processing and Materials，IPAM）全球公司的一个假想组织机构和IP Address Management（IPAM）（本书名）简写相同，有意思的文字游戏而已！）。IPAM全球公司的基本组织机构由在菲律宾的一个全球总部和分布于全球三个主要地理的分部组成，这三个分部分别位于欧洲的都柏林、北美的费城和亚洲的东京。IPAM全球公司有大约17000名雇员和24个配送中心（也作为分支办事处）和另外37个办事处（仅作为分支办事处）。图3-1给出一个基本的位置表格，突出显示了每个分部和相应的配送中心及分支办事处。

IP网络的部署将主要由各种因素决定，即用户所在的IP网络位于何处（依据图3-1中列出的站点）、在每个位置的用户数量、对信息资源（例如内部应用和因特网）访问的用户需求的多样性以及管理IP网络的行政管理（从安全到审计等）需求的多样性等所决定。因为与各种商务需要有关的输入的多样性，一般来说，任何一个机构的IP网络看起来与任何其他机构的IP网络都多少有点不同。但是，我们讨论的技术应该可以广泛地用于各种类型的网络（包括您的网络）。

IPAM全球公司的全球各个地点

核心站点	区域	地区站点	配送中心	分支办事处
费城	公司总部	费城		
费城	北美分部	费城		
	北美——东部	诺里斯敦	多伦多 纳舒厄 纽瓦克 巴尔的摩 匹兹堡 夏洛特 亚特兰大	普罗维登斯 昆西(Quincy) 奥尔巴尼 曼哈顿 海洋城 雷斯顿 里士满 查尔斯顿 蒙哥马利
	北美——中部	堪萨斯城	芝加哥 得梅因 孟菲斯 新奥尔良 墨西哥城	莱尔(Lisle) 印第安纳波利斯 托皮卡 休斯敦

图3-1　IPAM全球公司全球各处的位置和办事处

IPAM 全球公司的全球各个地点

核心站点	区域	地区站点	配送中心	分支办事处
	北美——西部	旧金山	丹佛 温哥华 菲尼克斯	卡尔加里 阿尔布开克 盐湖城 博尔德 埃德蒙顿 萨克拉门托 阿纳海姆
都柏林	欧洲分部	都柏林		
	欧洲——西部	伦敦	阿姆斯特丹 巴黎	曼彻斯特 满德里 里昂 里斯本
	欧洲——南部	罗马	罗马	尼斯 米兰 雅典
	欧洲——东部	柏林	慕尼黑 莫斯科	维也纳 布拉格 布达佩斯特 基辅
东京	亚洲分部	东京	东京 北京 新加坡 奥克兰	首尔 大阪 新加坡 马尼拉 新德里 悉尼

图 3-1　IPAM 全球公司全球各处的位置和办事处（续）

IPAM 全球公司的 IT 团队决定在机构总部和地区分部之间部署一个高速骨干或核心网。从每个区域分部办事处出发，为发射线状，形成的是一个大陆内部的广域网（WAN），将每个地区的零售点、配送点和分支办事处互联起来。依据这种基本的两层结构的核心网络和区域网络思路，构建而成的每个分支网络进一步分成地理区域。例如，在北美，分支网络将行政辖域分成三个子区域：东部、中部和西部，之后进一步分成主要的配送中心和分支办事处站点。类似地，欧洲区域分成西部、南部和东部区域。

依据这个拓扑，IT 团队决定在地址空间方面模仿这个结构，我们接下来可明白这点。因此，一个核心网络互联区域分部站点，每个区域分部作为其相应区域网络和核心网络之间的中继（intermediary）。每个区域网络将该区域内的相应配送中心和分支办事处互联起来。从一个机构角度看，每个区域有其自己的 IT 团队，该团队将负责管理其自己的空间以及关联的 DHCP 服务器与 DNS 服务器配置。图 3-2 画出高层的

IPAM 全球公司网络拓扑设计。

就 IP 地址空间分配而言，IPAM 全球公司将部署一个 10.0.0.0/8 网络，这是依据 RFC 1918 规定的私有地址空间。公开地址空间 192.0.2.0/24 是从一个 ISP 得到的（在本章后面，我们将讨论这个空间是从哪里得到的，并讨论 ISP 公开地址空间分配及其策略）。这个公开空间将分配给连接因特网的设备，如 Web 服务器、电子邮件网关和用于合作方连接和远端雇员的 VPN 网关。另外，该公开地址空间的一部分是保留部署的，保留用于连接 ISP 的一台网络地址转换（NAT）防火墙上的一个公开地址池。如我们在第 1 章介绍的，一台 NAT 可配置为自动地实施私有地址到公开地址的转换，其目的是使私有编址（内部）的主机能够访问因特网。

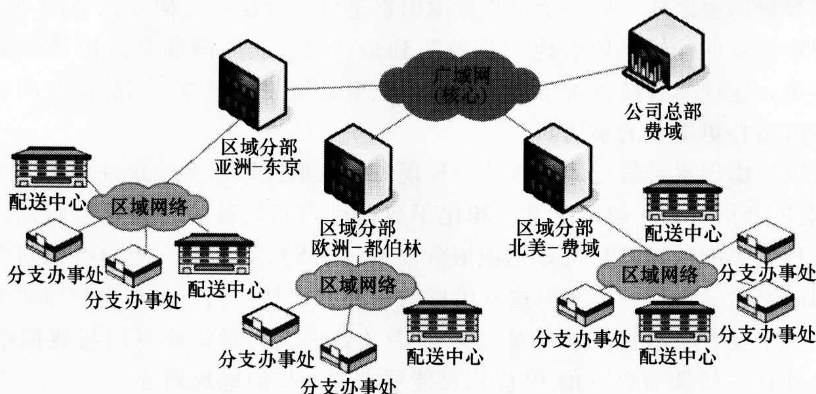


图 3-2　IPAM 全球公司的网络拓扑（部分拓扑）

3.1　地址分配逻辑[⊖]

有效的 IP 地址分配要求最佳的规划，理想情况下，需要准确的预测。了解网络层次结构每个层次的 IP 地址空间需求，使地址空间的最优分配可完全满足地址容量需要，同时最小化地址空间的浪费。当然在现实中，要得到一项准确的长期 IP 地址预测是极其罕见的奢求。商务需求驱动永恒的变化，新站点的开通，一些站点关闭了或迁走了，新的 IT 创业公司喜欢使用 IP 上的语音业务（IP 电话）办公，甚至洽谈兼并公司业务时也是如此。这些战略性事件通常可预先做出计划，除此之外，组织结构的动态变化在地址容量需求方面可能导致较短期的变化（紊乱）。例如，也许一个区域性的组织机构正在实施一项异常成功的顾客推广计划，导致 IP 地址需求的突然增加，或一个新项目正在导致 IP 地址需求的变动（原因是项目资源临时地处在相同位置）。

您的底线是在如下方面尽全力而为，即预先规划高层地址容量需求，为可能的规模扩展添加某个额外的“保障”（insurance）地址空间，之后提前监测地址使用率，

⊖　分配逻辑和范例的依据是参考文献［11］第 6 章中的类似内容。

以此作为一个反馈环路，确保在考虑短期和长期地址影响事件的情况下，所分配的地址正被有效地利用。作为 IP 空间管理中的一项主要功能，预先部署的监测可触发地址空间分配或移动到地址需要比较紧急之处。

当然，预先部署监测的密度将直接与地址空间的利用率成比例。如果您所在网络的利用率在 90%[⊖] 以上，那么您将需要每小时或至少一天数次监测它们的利用率。利用率在 70% 以下的网络，监测检查在一周数次的频度即可。理想情况是，在一个监测或 IP 地址管理系统内部，定义阈值和报警条件，从而缓解人工连续监测网络的需要，同时使关联的管理系统收集信息，并在出现一个特定的容量利用条件时向您报警。

除了容量需要之外，另一个重要考虑因素是以一种层次结构方式分配 IP 地址块，这样可使地址空间能够高效率地"折卷"到最高层次。为降低路由协议流量和路由表额外负担，这项实践措施对于最大化路由汇聚是至关重要的。当分配空间时，这是比考虑路由拓扑更重要的事情。

第三项考虑因素是最近才发生的一种现象：依据应用而实施地址分配。由于某些应用（例如 IP 语音，一般情况下，相比于数据可容忍数秒的时间延迟而言，它要求在数十毫秒量级的低延迟）的延迟或服务质量（QoS）需求，一些网络规划人员实施基于应用的路由处理措施。实施这种措施的一种方式是，例如分割出整体地址空间的一部分，用于具有较高优先级排队要求的语音处理，同时将此空间与数据空间隔离开。其他具有"特殊需要"的 IP 应用可能要求进一步的地址划分。

3.1.1 顶层分配逻辑

为了形象说明地址分配概念，让我们将这些概念应用到 IPAM 全球公司的私有地址块 10.0.0.0/8。当实施像这样的顶层分配时，要牢记在心的是，不仅以 IP 地址表示的所需容量是绝对必要的，而且子分块或层次结构各层的数量也是绝对必要的。在 IPAM 全球公司的情形中，我们将地址分层结构各层定义如下。

1）应用。
2）大陆级层或核心层。
3）区域层。
4）站点或建筑物。

因此，我们的顶层分配将依据应用分割的地址空间。那么将在核心路由器或大陆级层次分配每个应用特定的地址，之后依据区域，最后依据办事处进行地址分配。因为有四个层的分配结构，所以我们将不得不沿非字节边界进行分配。所以让我们从 CIDR 网络表示法和对应的二进制表示法两个角度对此进行考察。

这个网络的二进制表示如下所示。该地址的网络部分，其长度由/8 表示法加以

⊖ 采用占整块百分比的做法，并不总是可采取的最佳触发行动法，特别当在整个组织机构内部使用不同尺寸的子网时，更是如此。有 10 个地址的子网的 90% 利用率，当然比有 1000 个地址的子网的 90% 利用率要具有较高的地址需求紧迫度。

识别，以黑斜体突出显示，而本地部分是正常文本。

私有网络 10.0.0.0/8　***00001010*** 00000000 00000000 00000000

将这个空间在该机构各部分间分配时，让我们假定，IPAM 全球公司正计划在不久的将来要推广 IP 电话，并在以后推广其他 IP 服务。让我们取网络地址的下一个 4bit，并分配等尺寸的/12 网络。这将提供 $2^{(12-8)}=16$ 个潜在的高层应用，同时在每个分配中提供 $2^{(32-12)}>100$ 万个 IP 地址。因此，我们为每个基础设施地址空间分配一个/12 网络，为 IP 电话的 IP 地址“子空间”分配一个/12 网络，为数据子空间分配一个/12 网络。这种分配展示如下，黑斜体比特同样表示网络（网络 + 子网）部分，正常格式的比特代表主机比特。

私有网络　10.0.0.0/8　***00001010*** 00000000 00000000 00000000
基础设施　10.0.0.0/12　***00001010 0000***0000 00000000 00000000
语音　10.16.0.0/12　***00001010 0001***0000 00000000 00000000
数据　10.32.0.0/12　***00001010 0010***0000 00000000 00000000

3.1.2　第二层分配逻辑

这种比较初步的应用层分配法，将我们的最初一整块/8 地址空间重新规整为三个/12 空间（依据每个应用分配的）。在这个层次使用/12 的决定是第一层的分配数量和每个分配可用的地址数量之间的一个折中考虑。如果我们决定分配/11 形式的网络，那么我们将总的得到 8 个/11 网络，每个网络有 200 万以上的 IP 地址。在 IPAM 全球公司的情形中，考虑到每个/12 应用块有 100 万个 IP 地址，使未来分配有更多可用的顶层块，比起管理每块的容量来，是人们更加关注的问题。如果一个特定的分配用光了，则我们可分配另一个/12 块。

第二层次以及后续层次分配的块大小决策，一般来说，应该采用不同的逻辑。不应该在分配尺寸与相等尺寸分配数量之间的折中方面做文章，而应该使用一种优化的分配策略。这种优化策略需要不断地将地址空间缩小一半，直到达到要求的尺寸。采用这种方法的主要原因是，它使你能够保留较大块的未分配地址空间，以便用于较大的请求和替代分配。

如果您曾经经历过一个公司的合并，那么您可能遇到过如下的一种情景，这种情景很好地展示说明了优化分配的动机（出发点）。让我们假设 IPAM 全球公司收购了一家公司，网络集成策略要求向新的分部分配 250000 个 IP 地址。为了最小化混乱（并充分显示对此竞争 IT 组织机构的网络控制力），IPAM 全球公司期望分配单一的/14 网络，做到支持 262142 个地址。

如果我们优化地分配我们的地址空间，那么也许恰巧有一个/14 网络可用（IP 地址分配大师！）。如果在各处采用分配/16 网络的一种统一方法，也许幸运地得到可组成一个/14 网络的四个连续/16 网络（幸运的业余选手！）。如果没有找到四个连续的/16 网络，则不得不分配四个非连续的/16 网络；这会使路由表和路由协议更新表项增加 4 倍的额外负担（四个/16 网络对一个/14 网络——新手！）。采用连续的二分法，

而不是均匀单一尺寸的分割方法，更可能存在一个/14 网络可用于分配。让我们看看这是如何做到的。

如果我们从 10.0.0.0/12 基础设施地址块开始，并将之二分，得到如下所示的两个/13 地址块。利用二进制运算的特点，将下一个“主机”比特关联到网络，就会将原始网络二分。注意 10.0.0.0/12 网络不再存在，所以我们将其背景涂灰来说明这点；它已经被分成两个/13 网络。

原始网络	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
前一半	10.0.0.0/13	***00001010 00000***000 00000000 00000000
后一半	10.8.0.0/13	***00001010 00001***000 00000000 00000000

接下来，我们将上面的“前一半”二分，留下 10.8.0.0/13 地址块为未来用途或基础设施应用（或收购合并！）的子网划分。现在我们将地址的网络部分扩展到第 14bit，从而将 10.0.0.0/13 二分，得到如下的两个/14 网络。注意到这和处理 10.0.0.0/12 网络一样，10.0.0.0/13 网络也不再作为一个实体存在，所以也将其背景涂灰。它被分割成两个/14 网络，如下所示。但是，依据需要，该机构可将 10.8.0.0/13 网络用于进一步的分配。

原始网络	10.0.0.0/12	***00001010 0000***0000 00000000 00000000
原始网络的前一半	10.0.0.0/13	***00001010 00000***000 00000000 00000000
第一个/14	10.0.0.0/14	***00001010 000000***00 00000000 00000000
第二个/14	10.4.0.0/14	***00001010 000001***00 00000000 00000000
原始网络的后一半	10.8.0.0/13	***00001010 00001***000 00000000 00000000

从整体分配角度看，将这个二分过程可视化的一种方式是，将地址空间看作一个饼图，如图 3-3 所示。如果我们的整个大饼代表基础网络 10.0.0.0/12，那么我们将之二分，产生两个/13 网络，如图 3-3 左侧所示。之后，将一个/13 网络留作“可用”空间（左侧的一半），并将另一个/13 网络（右侧的一半）分成两个/14 网络，如图 3-3 右侧一半所示。

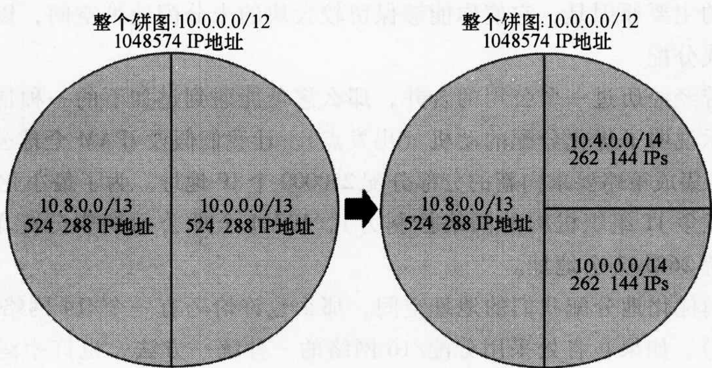


图 3-3 地址分配的饼图（依据参考文献 [11, 166]）

继续使用这个逻辑，直到得到一个/16 网络的情形，我们得到如下过程：

原始网络　10. 0. 0. 0/12 ***00001010 0000***0000 00000000 00000000

前一半(/13)　10. 0. 0. 0/13 ***00001010 00000***000 00000000 00000000

第一个/14　10. 0. 0. 0/14 ***00001010 000000***00 00000000 00000000

第一个/15　10. 0. 0. 0/15 ***00001010 0000000***0 00000000 00000000

第一个/16　10. 0. 0. 0/16 ***00001010 00000000*** 00000000 00000000

第二个/16　10. 1. 0. 0/16 ***00001010 00000001*** 00000000 00000000

第二个/15　10. 2. 0. 0/15 ***00001010 0000001***0 00000000 00000000

第一个/14　10. 4. 0. 0/14 ***00001010 000001***00 00000000 00000000

后一半(/13)　10. 8. 0. 0/13 ***00001010 00001***000 00000000 00000000

当每个“第一”地址块被分割时，它就创建了网络掩码长度比原始地址块长 1bit 的两个网络。既然我们执行了这种分割，则得到两个/16 网络：10. 0. 0. 0/16 和 10. 1. 0. 0/16，其产生过程如上所述。在两个突出显示的/16 地址块下面，也还有一个/15、一个/14 和一个/13 地址块可以使用。存在这些地址块的原因是，网络“第一个”集合被连续地分割成相等的两份，这样得到的“第一个”网络再进一步分割，“第二个”网络可保留为其他的未来分配或指派。最小的“第一个”网络，即 10. 0. 0. 0/16，是满足要求尺寸的、可分配的一个网络。

10. 0. 0. 0/16 ***00001010 00000000*** 00000000 00000000

10. 1. 0. 0/16 ***00001010 00000001*** 00000000 00000000

10. 2. 0. 0/15 ***00001010 0000001***0 00000000 00000000

10. 4. 0. 0/14 ***00001010 000001***00 00000000 00000000

10. 8. 0. 0/13 ***00001010 00001***000 00000000 00000000

但 IPAM 全球公司要求第三个/16 网络。我们应该从哪个地址块中分配这个网络呢？为了与保留较大的地址块的建议相一致，我们将取最小尺寸的下一个可用网络进行分配。在我们的情形中，从上面的列表看出，10. 2. 0. 0/15 网络可用于进一步的分配。如果我们将这个/15 网络分成两个/16 网络，则得到 10. 2. 0. 0/16 和 10. 3. 0. 0/16。之后我们将按照前面所展示说明的步骤，将这两个网络中的前一个网络加以分配，并将后一个网络用于未来分配。得到的饼图如图 3-4 所示，上面标有被分配的空间。

注意，我们仍然有许多大型地址块可用于进一步的分配或指派。仅有这个饼图中较暗阴影锲形部分组成已被分配的三个/16 网络。与上部的表中后续分割有关的饼图之中，在每个“第一个”二分之一地址块是被指派的或被分割成更小的分配，它得到一个对应的“第二个”二分之一地址块，该块仍然是空闲的或可用的。因此，依据这个初始分配，得到的 IPAM 全球公司

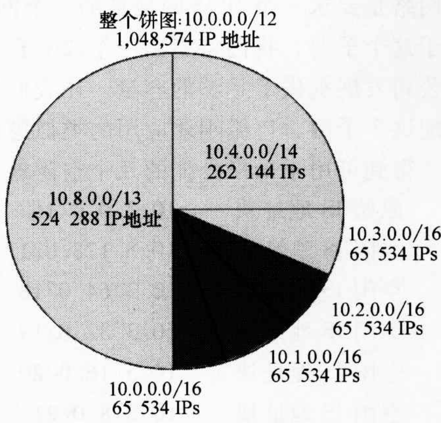


图 3-4　从/12 空间分配三个/16 网络空间（依据参考文献 [11，166]）

地址分配如下：

原始基础设施(IS)块 10.0.0.0/12 ***00001010 0000***0000 00000000 00000000

空闲的IS块 10.8.0.0/13 ***00001010 00001***000 00000000 00000000

空闲的IS块 10.4.0.0/14 ***00001010 000001***00 00000000 00000000

北美IS块 10.0.0.0/16 *00001010 00000000* 00000000 00000000

欧洲IS块 10.1.0.0/16 *00001010 00000001* 00000000 00000000

亚洲IS块 10.2.0.0/16 *00001010 00000010* 00000000 00000000

空闲IS块 10.3.0.0/16 ***00001010 00000011*** 00000000 00000000

对于数据和语音顶层地址分配 10.16.0.0/12 和 10.32.0.0/12 分别遵循类似逻辑，可推演得到如下分配：

原始语音应用的地址块 10.16.0.0/12 ***00001010 0001***0000 00000000 00000000

空闲的语音应用的地址块 10.24.0.0/13 ***00001010 00011***000 00000000 00000000

空闲的语音应用的地址块 10.20.0.0/14 ***00001010 000101***00 00000000 00000000

北美的语音应用的地址块 10.16.0.0/16 *00001010 00010100* 00000000 00000000

欧洲的语音应用的地址块 10.17.0.0/16 *00001010 00010001* 00000000 00000000

亚洲的语音应用的地址块 10.18.0.0/16 *00001010 00010010* 00000000 00000000

空闲的语音应用的地址块 10.19.0.0/16 ***00001010 00010011*** 00000000 00000000

原始数据应用的地址块 10.32.0.0/12 ***00001010 0010***0000 00000000 00000000

空闲的数据应用的地址块 10.40.0.0/13 ***00001010 00101***000 00000000 00000000

空闲的数据应用的地址块 10.36.0.0/14 ***00001010 001001***00 00000000 00000000

北美数据应用的地址块 10.32.0.0/16 *00001010 00100000* 00000000 00000000

欧洲数据应用的地址块 10.33.0.0/16 *00001010 00100001* 00000000 00000000

亚洲数据应用的地址块 10.34.0.0/16 *00001010 00100010* 00000000 00000000

空闲的数据应用的地址块 10.35.0.0/16 ***00001010 00100011*** 00000000 00000000

在这个核心层次剩下的唯一步骤是为核心路由器自身分配基础设施空间。毕竟核心网络是要求一个IP子网地址的一个网络，它位于我们的洲际间地址分配“之上”。对于这个子网，我们将划出一个/26子网。这个尺寸的子网提供62个主机地址，它为公司发展提供了足够的容量。让我们从最小的空闲基础设施地址块 10.3.0.0/16 中分配这个子网。遵循刚才应用的类似逻辑，我们向核心骨干网络分配 10.3.0.0/26 网络。得到可用于未来分配的几个空闲地址块：

原始IS地址块 10.3.0.0/16 ***00001010 00000011*** 00000000 00000000

空闲IS地址块 10.3.128.0/17 ***00001010 00000011 1***0000000 00000000

空闲IS地址块 10.3.64.0/18 ***00001010 00000011 01***000000 00000000

空闲IS地址块 10.3.32.0/19 ***00001010 00000011 001***00000 00000000

空闲IS地址块 10.3.16.0/20 ***00001010 00000011 0001***0000 00000000

空闲IS地址块 10.3.8.0/21 ***00001010 00000011 00001***000 00000000

空闲IS地址块 10.3.4.0/22 ***00001010 00000011 000001***00 00000000

空闲IS地址块 10.3.2.0/23 ***00001010 00000011 0000001***0 00000000

空闲 IS 地址块	10.3.1.0/24	***00001010 00000011 00000001*** 00000000
空闲 IS 地址块	10.3.0.128/25	***00001010 00000011 00000000 1***0000000
空闲 IS 地址块	10.3.0.64/26	***00001010 00000011 00000000 01***000000
核心网 IS 地址块	10.3.0.0/26	***00001010 00000011 00000000 00***000000

3.1.3　地址分配第 3 部分

既然在顶层依据应用分配了地址空间，那么在核心网络层次，每个这样的地址分配可被进一步划分，从而满足必要的配送中心和分支办事处的需求。本质上，这些地址分配用作相应区域内部为给定应用分配的地址块或地址池。这种自顶向下分配的技术可确保从这些初始分配的后续分配将可层次化地逐层折叠汇聚。因此，我们的核心路由器可简单地向其他核心路由器通告它们的/16 分配。同样，任何特定的依据服务报文的处理措施也可容易地进行配置。例如，如果想以最高优先级措施处理语音报文，那么我们可配置我们的路由器，对于以相应语音应用地址空间［例如用于欧洲语音流量的 10.17.0.0/16（或用于所有语音流量的 10.16.0.0/12）］的地址为源地址的报文提供这种处理。由此初始定义，在不影响这种处理逻辑的条件下，现在可沿地理的逻辑线逐步缩小区域的方式进一步分配。

让我们深入研究北美数据的地址空间 10.32.0.0/16。从前面图 3-1 给出的位置表中，看到北美站点被组织成三个区域：东部、中部和西部。我们也希望为各分部分配独立的地址空间。假定路由拓扑与这个地理机构是一致的，则我们将据此分配地址空间。因此，一个 WAN（广域网）将费城、堪萨斯城和旧金山的北美区域各站点与分部互联起来。这个区域型的互联结构代表了一个"亚核心"网络，和顶层结构一样，对此网络施用类似的分配逻辑。

让我们将 10.32.0.0/16 地址块划分成四个区域地址块。为了均匀地分配，需要将这个空间分成四个地址块。所以需要分配北美数据的地址空间中接下来的 2bit ($2^2=4$)，在如下的二进制表示中突出显示为较大字体的黑斜体比特。

北美数据	10.32.0.0/16	***00001010 00010000*** 00000000 00000000
北美分部数据	10.32.0.0/18	***00001010 00010000 00***000000 00000000
北美东部数据	10.32.64.0/18	***00001010 00010000 01***000000 00000000
北美西部数据	10.32.128.0/18	***00001010 00010000 10***000000 00000000
北美中部数据	10.32.192.0/18	***00001010 00010000 11***000000 00000000

虽然这会得到相等尺寸的分配，但我们不必按照 2 的幂次进行分配（依据我们所说明的情形）。我们刚才容易地将一个较大地址部分分配给了东部，原因是它包含了最多的站点：10.32.0.0/17（东部）、10.32.160.0/19（中部）[㊀]、10.32.192.0/19（西部）和 10.32.220.0/19［分部（HQ）］。

从这点看来，依据寻址需要，我们可从每个区域的地址空间中向其相应的站点分配地址。考虑北美西部数据的地址空间 10.32.128.0/18，现在可为每个配送中心和分

㊀ 原书为"10.32.160.0.0/19"，多了一个"0"。——译者注

支办事处的数据应用分配空间。这种分配的最简单策略是均匀分布，例如，就像我们在顶部分配层次所实施的方法一样，为每个站点分配相同尺寸的地址块。但是，人们需要考虑每个站点的用户数量和数据设备数量，规划每个站点的增长，在区域内部规划新的站点，同时考虑应用联网的需求。在 IPAM 全球公司的案例中，典型情况下，各配送中心安置有 65 名雇员，还有其他自动化机器和基础设施，总共需要 IP 地址约 200～250 个。分支办事处仅需要约 150～200 个 IP 地址，包括有关的笔记本计算机、PDA 以及其他数据设备，平均为 40 名雇员所用。

在这样一个场景中，为配送中心至少分配一个/23 网络（可提供 510 个可用 IP 地址）和为分支办事处分配一个/24 网络（提供 254 个 IP 地址）是合理的。但是，考虑到每个站点相应的寻址需求，应该分别对每个站点进行分析。在我们的情形中，首先要为每个配送中心分配一个/23 网络，之后为每个分支办事处分配一个/24 网络。如下所示给出这个分配，还有源于原始 10.32.128.0/18 网络的剩余空闲空间，这可用于未来的分配。

北美西部数据	10.32.128.0/18	***00001010 00100000 10***000000 00000000
旧金山站点	10.32.128.0/23	***00001010 00100000 1000000***0 00000000
丹佛站点	10.32.130.0/23	***00001010 00100000 1000001***0 00000000
温哥华站点	10.32.132.0/23	***00001010 00100000 1000010***0 00000000
菲尼克斯站点	10.32.134.0/23	***00001010 00100000 1000011***0 00000000
卡尔加里站点	10.32.136.0/24	***00001010 00100000 10001000*** 00000000
阿尔布开克站点	10.32.137.0/24	***00001010 00100000 10001001*** 00000000
盐湖城站点	10.32.138.0/24	***00001010 00100000 10001010*** 00000000
博尔德站点	10.32.139.0/24	***00001010 00100000 10001011*** 00000000
埃德蒙顿站点	10.32.140.0/24	***00001010 00100000 10001100*** 00000000
萨克拉门托站点	10.32.141.0/24	***00001010 00100000 10001101*** 00000000
阿纳海姆站点	10.32.142.0/24	***00001010 00100000 10001110*** 00000000
空闲空间	10.32.143.0/24	***00001010 00100000 10001111*** 00000000
空闲空间	10.32.144.0/20	***00001010 00100000 1001***0000 00000000
空闲空间	10.32.160.0/19	***00001010 00100000 101***00000 00000000

针对总部位置，我们为每个主要的公司分部分配/22 网络。依据联网部署情况，这些分配可进一步划分子网。

3.1.4 分配均衡和跟踪

随着在地址分配层次结构中添加较多层次，地址的网络部分就会增长，同时就缩减了可指派给 IP 设备的主机所用比特的数量。前面表中列出的每个站点都有 8 或 9 个可用主机比特，分别为每个站点提供了 254 或 510 个独立的 IP 主机容量。结构化的分层法使将地址空间映射到应用、区域并最终到子网的做法成为可能，有助于保留地址汇总（对应于路由器拓扑和部署）。考虑在每个站点需要多少个 IP 地址容量，并将此容量与期望多少结构层次综合考虑，这种做法是不错的想法。

每个子网的独立 IP 地址容量需求，将有助于您得到端点[⊖]的分配尺寸。许多组织机构计划为每个端子网在一个/24 分配中分配 254 台主机（地址）。如果需要可分配多个子网。使用这个字节边界的做法，有助于简化从二进制到十进制转换的工作（如上面小节中看到的情况），但由于地址容量需求，这种做法对您的组织机构可能是行不通的。如果您需要越过字节边界进行分配，则使用一个 IP 地址管理工具也许有助于确保分配的准确性（在没有重叠的情况下），同时保持了地址的层次结构。

无论您是否决定使用一个 IP 管理系统，都必须跟踪记录地址分配的情况。为了形象地说明一种简单的跟踪记录方法，我们重写了本章开始处给出的表格，该表格列出了 IPAM 全球公司的网络位置，它反映了相应的地址块分配。在下面给出的图 3-5 所示的更新版中，在同一站点列中列出了配送中心和分支办事处，在前边以一种浅阴影字体列出配送中心。

针对每个区域，在每个结构层次组成地址池（供应）的顶层结构地址块是突出显示的，为的是将它们与子网做出区分。为了使这件事做起来简单，我们遵循一种常用分配方法，为每个配送中心分配一个/23 网络，为每个分支办事处分配一个/24 网络。我们仅形象说明了表格中给出的一个小型子网，但相同的方法论可用于欧洲站点和亚洲站点，同样适用于语音和数据应用。

这种分配形式的一个便捷的作用是得到这样一种能力，即容易地将一个地址与一个位置关联起来。例如，知道 10.0.79.0/24 是奥尔巴尼的基础设施子网，那么人们就可推断 10.16.79.0/24 是 VoIP 子网，10.32.79.0/24 是数据子网。一眼就可看出，10.*X*.*Y*.0 网络的这种字节模式将应用（字节 *X*）和位置（字节 *Y*）映射关联起来。在我们的范例中，字节 *X* 为 0 指基础设施，16 指 VoIP，32 指数据。在这个范例中，字节 *Y* 为 79 指奥尔巴尼。

区域	区域的站点	站点	基础设施网	VoIP 网	数据网
公司总部	费城		10.0.0.0/12	10.16.0.0/12	10.32.0.0/12
北美分部	费城		10.0.0.0/12	10.16.0.0/12	10.32.0.0/12
		核心网	10.3.0.0/26		
		费城——行政	10.0.0.0/22	10.16.0.0/22	10.32.0.0/22
		费城——财务	10.0.4.0/22	10.16.4.0/22	10.32.4.0/22
		费城——运行	10.0.8.0/22	10.16.8.0/22	10.32.8.0/22
		费城——技术	10.0.12.0/22	10.16.12.0/22	10.32.12.0/22
		费城——营销	10.0.16.0/22	10.16.16.0/22	10.32.16.0/22
		费城——研发	10.0.20.0/22	10.16.20.0/22	10.32.20.0/22
北美——东部	诺里斯敦		10.0.64.0/18	10.16.64.0/18	10.32.64.0/18
		诺里斯敦	10.0.64.0/23	10.16.64.0/23	10.32.64.0/23

图 3-5 IPAM 全球公司的 IPv4 地址块分配（部分分配）

⊖ Endpoint：端点，即桩网络。——译者注

区域	区域的站点	站点	基础设施网	VoIP 网	数据网
		多伦多	10.0.66.0/23	10.16.66.0/23	10.32.66.0/23
		纳舒厄	10.0.68.0/23	10.16.68.0/23	10.32.68.0/23
		纽瓦克	10.0.70.0/23	10.16.70.0/23	10.32.70.0/23
		巴尔的摩	10.0.72.0/23	10.16.72.0/23	10.32.72.0/23
		匹兹堡	10.0.74.0/23	10.16.74.0/23	10.32.74.0/23
		夏洛特	10.0.76.0/23	10.16.76.0/23	10.32.76.0/23
		亚特兰大	10.0.77.0/24	10.16.77.0/24	10.32.77.0/24
		普罗维登斯	10.0.78.0/24	10.16.78.0/24	10.32.78.0/24
		昆西	10.0.79.0/24	10.16.79.0/24	10.32.79.0/24
		奥尔巴尼	10.0.80.0/24	10.16.80.0/24	10.32.80.0/24
		曼哈顿	10.0.81.0/24	10.16.81.0/24	10.32.81.0/24
		海洋城	10.0.82.0/24	10.16.82.0/24	10.32.82.0/24
		雷斯顿	10.0.83.0/24	10.16.83.0/24	10.32.83.0/24
		里士满	10.0.84.0/24	10.16.84.0/24	10.32.84.0/24
		查尔斯顿	10.0.85.0/24	10.16.85.0/24	10.32.85.0/24
		蒙哥马利	10.0.86.0/24	10.16.86.0/24	10.32.86.0/24
北美——中部	堪萨斯城		10.0.192.0/18	10.16.192.0/18	10.32.192.0/18
		堪萨斯城	10.0.192.0/23	10.16.192.0/23	10.32.192.0/23
		芝加哥	10.0.194.0/23	10.16.194.0/23	10.32.194.0/23
		得梅因	10.0.196.0/23	10.16.196.0/23	10.32.196.0/23
		孟菲斯	10.0.198.0/23	10.16.198.0/23	10.32.198.0/23
		新奥尔良	10.0.200.0/23	10.16.200.0/23	10.32.200.0/23
		……	……	……	……

图 3-5 IPAM 全球公司的 IPv4 地址块分配（部分分配）（续）

3.1.5 IPAM 全球公司的公开地址空间

现在让我们看看从 ISP 得到的 IPAM 全球公司的公开地址空间 192.0.2.0/24。在本章后面我们将讨论 ISP 用来得到 IP 地址空间的过程。IPAM 全球公司在费城总部办事处有一条到所选择 ISP 的因特网连接。虽然两条不同路由的本地环路提供一定程度的接入冗余性，但未来计划要求从另一个地址提供对多穴连接的支持，我们也将在稍后讨论这点。目前在/24 网络内部的 254 个可用公开 IP 地址，将用于寻址因特网（外部）可达的主机（例如 web 和电子邮件服务器），一个共享的地址池使内部客户能够访问因特网。安装了一对 NAT 设备，支持负载均衡共享和地址转换，用于内部客户访问因特网。事实上，这个/24 网络将可能需要划分子网，将因特网可达主机和 NAT 地址分隔开来。

3.2 IPv6 地址分配[㊀]

虽然 IPv6 地址的表示不同于 IPv4 地址，但从本质上而言，分配过程的工作原理是相同的。主要区别在于 IPv6 十六进制和二进制之间的相互转换，IPv4 则是十进制和二进制之间的相互转换。上述针对 IPv4 的最小可用空闲地址块的最优指派过程，是最佳拟合分配算法的一个范例。由于可用地址空间的巨大差异，IPv6 不仅支持一个类似的最佳拟合算法，而且支持一个稀疏分配方法。我们也讨论一种随机分配方法，可用来替换从 1 开始逐步增加的简单子网地址分配法。

在本节我们使用范例 IPv6 网络 2001：DB8：：/32，概要描述这里所说的每个算法。注意，/32（或任何）——尺寸大小的全局单播分配，都要求向一个区域因特网注册权威机构（RIR）提前认证申请，我们将在本章后面讨论这点，像 IPAM 全球公司这样的一个组织机构不太可能接受这样的一种分配。但是，最初在我们的范例中将使用这样的一种分配，这样使比特数不会超出页面（即显示不了）。后面我们将使用一个比较实际的/48 范例分配。对于以/32 或/48 开始的算法将是等价的，对于/48 网络而言，仅是有更多的中间 0 前缀比特而已。

3.2.1 最佳拟合分配

使用一种最佳拟合方法，我们将遵循前面针对 IPv4 描述的同样的基本按照比特分配的算法。在将十六进制转换为二进制之后，就后续的二分法，即将地址的下一比特用于网络部分的方法而言，这个过程是完全相同的。例如，下面考虑我们的范例网络 2001：0DB8：：/32。

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000 0000…

假定，我们将从这个空间分配三个/40 网络。从二进制角度来看，遵循类似 IPv4 分配范例的方法，后续地将地址空间逐步二分到一个/40 大小，如下较大黑斜体比特所示，您应该得到如下信息：

001000000000000100001101101110001000 0000000000000000...

00100000000000010000110110111000 0100 0000000000000000...

00100000000000010000110110111000 00100000000000000000...

00100000000000010000110110111000 00010000000000000000...

00100000000000010000110110111000 00001000000000000000...

00100000000000010000110110111000 00000100000000000000...

00100000000000010000110110111000 00000010000000000000...

00100000000000010000110110111000 00000001000000000000...

00100000000000010000110110111000 00000000000000000000...

这里我们已经有两个/40 网络可用（上面突出显示的），并将这两个网络转换为

㊀ 此处 IPv6 分配的讨论依据参考文献［172］。

十六进制，则我们得到 2001：0DB8：0100：：/40 和 2001：0DB8：0000：/40（即 2001：DB8：：/40）。在此分配之后，使用最佳拟合方法来分配第三个/40 网络，则可取下一个最小的可用网络，在这种情形中是一个/39 网络，将其分成两个/40 网络：

0010 0000 0000 0001 0000 1101 1011 1000 0000 0010 0000 0000 0000

0010 0000 0000 0001 0000 1101 1011 1000 0000 0011 0000 0000 0000

通过取下一个比特，将这个网络二分，得到两个/40 网络。我们可选择一个网络加以分配，另外一个网络将用于未来的地址指派。所以我们分配的三个/40 网络是 2001：DB8：：/40、2001：DB8：0100：：/40 和 2001：DB8：0200：：/40。另一个/40 网络，即 2001：DB8：0300/40 可用于未来的地址分配。图 3-6 以一种饼图形式形象化地说明了这个连续的二分过程。

在分配这三个/40 网络之后，如图 3-6 中突出显示的，饼图的剩余部分可用于（未来的）地址分配。在上面列出的连续二分过程中，这些可用网络作为顶部六项出现，还有前一个网络 2001：DB8：200：：/39 未分配的前半部分。

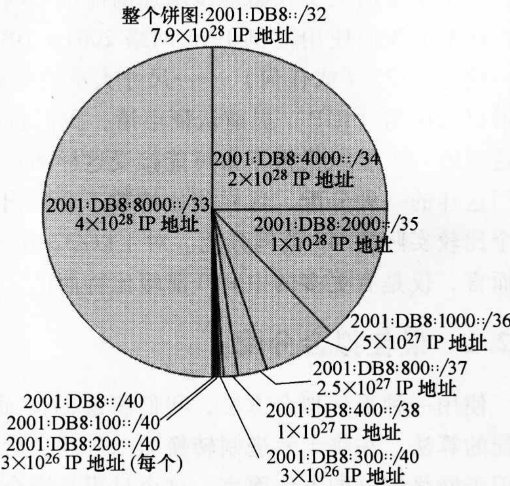


图 3-6 从一个/32 网络切分出三个/40 网络的分配结果（依据参考文献［11］和［166］）

3.2.2 稀疏分配方法

从以前的算法中注意到，通过从一个/32 网络分配一个/40 网络的方法，我们递增地将网络长度扩展到第 40bit，这和 IPv4 分配的过程相同。之后，我们将一个 0 或 1 指派给第 40bit，而将该网络指派为我们的前两个/40 网络。本质上而言，以此法处理每个比特，考虑“1”代表空闲地址块，“0”代表已分配的地址块。但是，如果我们退一步看，将 8 个子网 ID 比特（将/32 扩展到/40）看作一个整体，这种做法并不是递增的二分网络法，观察到我们实际上分配了子网，做法是在子网 ID 字段内部简单地编址或计数而已，表示为这个表中的突出显示黑斜体 bit。

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000

0000. . . 2001：DB8：：/40

0010 0000 0000 0001 0000 1101 1011 1000 0000 0001 0000 0000

0000. . . 2001：DB8：100：：/40

0010 0000 0000 0001 0000 1101 1011 1000 0000 0010 0000 0000

0000. . . 2001：DB8：200：：/40

因此，如果事先知道原始的/32 网络将被均匀地分割为唯一的/40 大小的地址块，那么一种比较简单的分配方法将是简单地递增子网 ID 比特（即每次加 1 操作）。接下

来/40 网络的分配将使用子网 ID 数值 00000011、00000100、00000101 等。在一些网络中，分配/40 地址块的这种均匀法策略可能并不适用，所以连续二分的方法可能是比较合适的。

另一方面，如果您所在的机构是一个本地因特网注册机构（LIR）或 ISP，则一种稀疏分配方法也许是有吸引力的。稀疏分配方法寻求将地址分配分散化，采用在地址分配之间分配最大空间的方法，为增长留下空间。稀疏算法的特征也是将可用地址空间进行二分，但它并不将这个过程继续到最小地址尺寸，它要求在新的一半地址空间的边缘分配下一个地址块。这导致地址分配分散化，且不是最优分配的。同样，其哲学依据是，通过在充足的 IPv6 空间各分配之间留下丰富的空间，这为所分配网络的发展提供了空间。考虑一个范例，从 2001：DB8：：/32 空间分配三个/40 网络，看起来的情形如下：

0010 0000 0000 0001 0000 1101 1011 1000 0000 0000 0000 0000

0000. . . 2001：DB8：：/40

0010 0000 0000 0001 0000 1101 1011 1000 1000 0000 0000 0000

0000. . . 2001：DB8：800：：/40

0010 0000 0000 0001 0000 1101 1011 1000 0100 0000 0000 0000

0000. . . 2001：DB8：4000：：/40

这些二进制表示分别转换为 2001：DB8：：/40、2001：DB8：8000：：/40 和 2001：DB8：4000：：/40。这种分配使地址空间分散化，如图 3-7 所示。如果 2001：DB8：8000：：/40 网络的接受者要求附加的地址分配，则可分配一个连续的或邻接的地址块 2001：DB8：8100：：/40。采用稀疏方法时，这个地址块将在所分配块的最后一块，所以它可用的概率较大。在这样一种情形中，两个连续地址块的接受者可将他们的地址空间标识（并通告）为 2001：DB8：8000：：/39。注意子网 ID 比特实际上是从左到右计数的，而不采用“正常”计数的常规从右向左的方法。

RFC 3531[22] 描述了稀疏分配方法论。因为人们期望网络分配遵循一个多层的分配层次结构，所以不同实体进行连续分配时，可使用几组连续的网络比特。例如，一个因特网注册机构可向一个区域注册机构分配第一层次的地址宏块，该区域注册机构接下来从那个空间向一个服务提供商分配地址块，该服务提供商接着从那个子空间向其顾客分配地址块，顾客可进一步在其各网络间分配地址块。RFC 3531 建议，较高层次的地址分配（例如各注册机构的地址分配）利用最左侧计数或稀疏分配法，最低层次的地址分配使用最右侧或最佳拟合分配法，处于中间的其他地址分配可使用上述任何一种分配法，或甚至使用一种从中间开始的分配方案。对于像 IPAM 全球公司这样的一个组织机构，使用稀疏方法分配洲际网络，在顶层为未来发展留下了空间。注意虽然 RFC 3531 解决的是 IPv6 地址分配问题，但也可以这种方式分配 IPAM 全球公司的顶层 IPv4 空间，从而将初始分配分散为基础设施的 10. 0. 0. 0/12、VoIP 的 10. 128. 0. 0/12 和数据的 10. 64. 0. 0/12。

3. 2. 3　随机分配

随机分配方法选择子网比特范围内部的一个随机数来分配子网。使用来自一个/

32 网络的/40 分配为例，将产生 0 和 2^8-1（或 255）之间的一个随机数，并在假定该数仍然可用的情况下实施分配。这种方法为在所分配实体间随机分散地址分配提供了一种方法，一般来说，对于“相同大小”的地址分配而言，效果最好。在不以从“1”开始连续地分配有序的地址块和子网方面，随机化提供了“隐私性”的一个层面。要小心的是，随机分配会使较大型连续地址块的识别（依据我们前面的合并范例）和为重新编址目的而释放连续空间这两方面更加困难。所以在顶层地址分配采用稀疏分配是有道理的，而在子网分配层次，随机或最佳拟合方法是比较合适的。

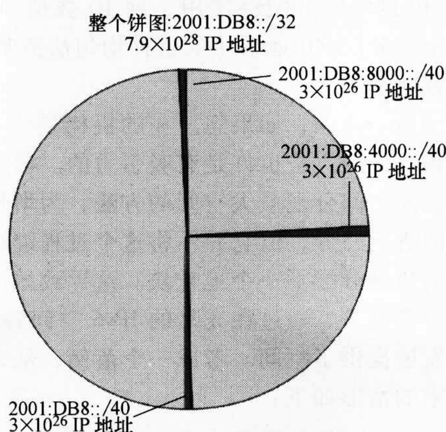


图 3-7　稀疏分配范例
（依据参考文献［11，166］）

3.2.4　唯一的本地地址空间

虽然 IPv6 没有指定“私有的”地址空间，但唯一本地地址（ULA）空间的概念本质上是等价于“私有”空间的。通过使用 FC00::/7 前缀，将 *L*bit 设置为“1”（即 FD00::/8），依此表明本地指派，并指派一个随机的 40bit 全局 ID，那么 IPAM 全球公司就为内部用途构造了一个/48 网络。在可能支持区域因特网注册结构的因特网共同体内部，为分配全球唯一的 40bit 全局 ID 存在某种讨论，区域因特网注册机构的职责是分配公开 IP 地址空间（接下来我们将看到是如何分配的）。尽管是全球唯一的，以 ULA 为目的地址的报文也不应该路由到一个组织机构的外部。但是，这项规定的强制实施，一般来说由每个组织结构执行，从而限制这样的报文越过它的外部边界路由器。

3.3　IPAM 全球公司的 IPv6 地址分配

虽然当描述地址分配时，出于说明的目的，我们使用了一个/32 网络，现在针对 IPAM 全球公司的范例，让我们使用一个比较真实的/48 大小地址块：2001：DB8：4AF0::/48。同样，虽然 IPAM 全球公司具有充足的公开 IPv6 空间，出于范例的目的，让我们也选择一个 ULA 网络加以分配：FD01：273E：90A::/48。将以与 IPAM 全球公司地理结构一致的方式，层次化地分配这些地址块。将和在前面的 10.*X*.*Y*.0 范例中的做法一样，出于模式一致性和较容易地实现可视化相关关系，使用针对每个位置的通用子网 ID 号，来分配这些网络。

在 IPAM 全球公司的情况下，让我们在核心网络层使用稀疏分配。从这个分配中，可进一步稀疏地分配到我们的各区域，之后对于我们的配送中心和分支办事处使用一种最佳拟合方法。虽然当前在 IPv6 之上我们仅运行数据应用，但仍然应该实施

一种应用层次的地址分配，以便应对应用扩展或未来增长。

出于简单性考虑，我们将在 4bit 边界上进行地址分配[㊀]。如果我们使用第一个 4 子网比特（49 ~ 52bit），将有 16 种可能的分配。因为到目前为止我们只有一个应用，所以针对“数据”应用，将简单地分配四个网络，使其代表核心网络。使用稀疏方法，得到如下地址分配。

核心分配	49 ~ 52bit	公开地址空间分配	ULA 地址分配
总部	**0 0 0 0**	2001：DB8：4AF0：：/52	FD01：273E：90A：：/52
北美	**1 0 0 0**	2001：DB8：4AF0：8000：：/52	FD01：273E：90A：8000：：/52
欧洲	**0 1 0 0**	2001：DB8：4AF0：4000：：/52	FD01：273E：90A：4000：：/52
亚洲	**1 1 0 0**	2001：DB8：4AF0：C000：：/52	FD01：273E：90A：C000：：/52

应用一种类似的方法，使用 53 ~ 56bit 作为下一层次分配，得到如下子（次级）分配。

次级核心分配	53 ~ 56bit	公开地址空间分配	ULA 分配
北美——东部	**0 0 0 0**	2001：DB8：4AF0：8000：：/56	FD01：273E：90A：：/56
北美——中部	**1 0 0 0**	2001：DB8：4AF0：8800：：/56	FD01：273E：90A：8800：：/56
北美——西部	**0 1 0 0**	2001：DB8：4AF0：8400：：/56	FD01：273E：90A：8400：：/56
欧洲——西部	**0 0 0 0**	2001：DB8：4AF0：4000：：/56	FD01：273E：90A：4000：：/56
欧洲——南部	**1 0 0 0**	2001：DB8：4AF0：4800：：/56	FD01：273E：90A：4800：：/56
欧洲——东部	**0 1 0 0**	2001：DB8：4AF0：4400：：/56	FD01：273E：90A：4400：：/56

在每个这样的/56 地址分配内部，可进一步为每个配送中心和分支办事处分配独立的/64 子网地址。我们将使用一种最佳拟合方法实施这种分配，并在扩展地址分配表格中汇总这种分配的一个子集。一般而言，IPv6 子网应该采用/64 网络前缀进行分配。许多 IPv6 特色功能（例如邻居发现）假定采用（依赖于）这个前缀长度。

对于路由器点到点链路或背靠背链路，可指派一个/126 子网，这类似于 IPv4 中提供两台主机地址的一个/30 网络。但是，要小心在 IPv6 地址的接口标识符字段内部“*u*”（全局/本地，第 71bit）和“*g*”（个体/组，第 72bit）这两个比特的设置。不正确地设置这些比特会影响访问或利用它们的各项应用。“*u*” bit 表明公司 ID 是由 IEEE（1）指派或本地（0）指派的，“*g*” bit 表明该地址是一个单播（0）或一个组播（1）地址。不应该使用/127 地址。/128 前缀指示单一 IP 地址，类似于 IPv4 中的/32。

让我们将这些 IPv6 地址分配添加到 IPAM 全球公司的 IP 地址表格，如下所示。

核心站点	区域	区域的站点	站点	基础设施网	VoIP 网	数据网	公开 IPv6 地址	IPv6 ULA
费城	公司总部	费城		**10.0.0.0/12**	**10.16.0.0/12**	**10.32.0.0/12**	**2001:DB8:4AF0::/52**	**2001:273E:90A::/52**

㊀ 在第 9 章，我们将讨论非 4bit 边界上进行地址分配的隐含意义，将特别讨论对 DNS 的隐含意义。

（续）

核心站点	区域	区域的站点	站点	基础设施网	VoIP网	数据网	公开IPv6地址	IPv6 ULA
费城	北美分部	费城		10.0.0.0/16	10.16.0.0/16	10.32.0.0/16	**2001:DB8:4AF0::/56**	**2001:273E:90A:8000::/52**
			核心网	10.3.0.0/26			2001:DB8:4AF0:800::/64	2001:273E:90A:800::/64
			费城——行政	10.0.0.0/22	10.16.0.0/22	10.32.0.0/22	2001:DB8:4AF0::/64	2001:273E:90A::/64
			费城——财务	10.0.4.0/22	10.16.4.0/22	10.32.4.0/22	2001:DB8:4AF0:1::/64	2001:273E:90A:1::/64
			费城——运行	10.0.8.0/22	10.16.8.0/22	10.32.8.0/22	2001:DB8:4AF0:2::/64	2001:273E:90A:2::/64
			费城——技术	10.0.12.0/22	10.16.12.0/22	10.32.12.0/22	2001:DB8:4AF0:3::/64	2001:273E:90A:3::/64
			费城——营销	10.0.16.0/22	10.16.16.0/22	10.32.16.0/22	2001:DB8:4AF0:4::/64	2001:273E:90A:4::/64
			费城——研发	10.0.20.0/22	10.16.20.0/22	10.32.20.0/22	2001:DB8:4AF0:5::/64	2001:273E:90A:5::/64
	北美——东部	诺里斯敦		**10.0.64.0/18**	**10.16.64.0/18**	**10.32.64.0/18**	**2001:DB8:4AF0:8000::/56**	**2001:273E:90A:8000::/56**
			诺里斯敦	10.0.64.0/23	10.16.64.0/23	10.32.64.0/23	2001:DB8:4AF0:8000::/64	2001:273E:90A:8000::/64
			多伦多	10.0.66.0/23	10.16.66.0/23	10.32.66.0/23	2001:DB8:4AF0:8001::/64	2001:273E:90A:8001::/64
			纳舒厄	10.0.68.0/23	10.16.68.0/23	10.32.68.0/23	2001:DB8:4AF0:8002::/64	2001:273E:90A:8002::/64
			纽瓦克	10.0.70.0/23	10.16.70.0/23	10.32.70.0/23	2001:DB8:4AF0:8003::/64	2001:273E:90A:8003::/64
			巴尔的摩	10.0.72.0/23	10.16.72.0/23	10.32.72.0/23	2001:DB8:4AF0:8004::/64	2001:273E:90A:8004::/64

（续）

核心站点	区域	区域的站点	站点	基础设施网	VoIP 网	数据网	公开 IPv6 地址	IPv6 ULA
			匹兹堡	10.0.74.0/23	10.16.74.0/23	10.32.74.0/23	2001:DB8:4AF0:8005::/64	2001:273E:90A:8005::/64
			夏洛特	10.0.76.0/23	10.16.76.0/23	10.32.76.0/23	2001:DB8:4AF0:8006::/64	2001:273E:90A:8006::/64
			亚特兰大	10.0.77.0/24	10.16.77.0/24	10.32.77.0/24	2001:DB8:4AF0:8007::/64	2001:273E:90A:8007::/64
			普罗维登斯	10.0.78.0/24	10.16.78.0/24	10.32.78.0/24	2001:DB8:4AF0:8008::/64	2001:273E:90A:8008::/64
			昆西	10.0.79.0/24	10.16.79.0/24	10.32.79.0/24	2001:DB8:4AF0:8009::/64	2001:273E:90A:8009::/64
			奥尔巴尼	10.0.80.0/24	10.16.80.0/24	10.32.80.0/24	2001:DB8:4AF0:800A::/64	2001:273E:90A:800A::/64
			曼哈顿	10.0.81.0/24	10.16.81.0/24	10.32.81.0/24	2001:DB8:4AF0:800B::/64	2001:273E:90A:800B::/64
			海洋城	10.0.82.0/24	10.16.82.0/24	10.32.82.0/24	2001:DB8:4AF0:800C::/64	2001:273E:90A:800C::/64
			雷斯顿	10.0.83.0/24	10.16.83.0/24	10.32.83.0/24	2001:DB8:4AF0:800D::/64	2001:273E:90A:800D::/64
			里士满	10.0.84.0/24	10.16.84.0/24	10.32.84.0/24	2001:DB8:4AF0:800E::/64	2001:273E:90A:800E::/64
			查尔斯顿	10.0.85.0/24	10.16.85.0/24	10.32.85.0/24	2001:DB8:4AF0:800F::/64	2001:273E:90A:800F::/64
			蒙哥马利	10.0.86.0/24	10.16.86.0/24	10.32.86.0/24	2001:DB8:4AF0:8010::/64	2001:273E:90A:8010::/64
北美——中部		堪萨斯城		**10.0.192.0/18**	**10.16.192.0/18**	**10.32.192.0/18**	**2001:DB8:4AF0:8800::/56**	**2001:273E:90A:8800::/56**
			堪萨斯城	10.0.192.0/23	10.16.192.0/23	10.32.192.0/23	2001:DB8:4AF0:8800::/64	2001:273E:90A:8800::/64
			芝加哥	10.0.194.0/23	10.16.194.0/23	10.32.194.0/23	2001:DB8:4AF0:8800::/64	2001:273E:90A:8800::/64
			得梅因	10.0.196.0/23	10.16.196.0/23	10.32.196.0/23	2001:DB8:4AF0:8801::/64	2001:273E:90A:8801::/64
			孟菲斯	10.0.198.0/23	10.16.198.0/23	10.32.198.0/23	2001:DB8:4AF0:8802::/64	2001:273E:90A:8802::/64
			新奥尔良	10.0.200.0/23	10.16.200.0/23	10.32.200.0/23	2001:DB8:4AF0:8803::/64	2001:273E:90A:8803::/64

如果保持这种增长，那么将需要更多页来记录这些信息。如我们提到的 IPv4 分配一样，比如匹茨堡使用“站点号”73（第三个字节），我们可识别它的 IPv6 站点号为 8005，这是第四个冒号段。从我们的公开 IPv6 地址空间中，将为我们的外部（因特网）可访问的服务器（例如 DNS、web、文件传输和电子邮件服务器）分配另外一个地址分配。使用两种不同的地址空间分配方法来分配 IPv4 空间：针对内部分配的私有地址空间法和针对外部访问的公开地址空间法。对于 IPv6，分配了公开地址空间和内部的 ULA 地址空间，还需要为外部可访问能力增加一个地址分配。让我们分配 2001：DB8：4AF0：2000：：/56 网络指派给外部主机。

现在完成了 IPAM 全球公司初始分配计划，并在表格中记录了每个地址分配。让我们回过头来，讨论一下公开 IP 地址空间是如何管理并分配给各 ISP 的，之后比较详细地描述多穴（使用多个 ISP）接入法。

3.4 因特网注册机构

为了做到正确路由和通信，在一个给定网络上，IP 地址必须是唯一的[⊖]。在全球因特网上如何确保这种唯一性呢？因特网地址分配号码权威机构（IANA）负责 IPv4 和 IPv6 的全球 IP 地址空间分配，同时负责 TCP/IP 内部所用其他参数的分配（例如应用端口号）。事实上，通过浏览 www. iana. org，选择号码资源链接下的“因特网协议 v4 地址空间”（Internet Protocol v4 Address Space）或“IPv6 地址空间”（IPv6 Address Space），您可查看这些顶层分配（23）。

从本质上而言，IANA 是顶层地址注册管理机构，它将地址空间分配给区域因特网注册机构（RIR）。如下面列出的，RIR 是这样的组织机构，它们负责在其相应的全球区域内部，从其对应的地址空间资源中（从 IANA 得到）实施地址空间分配。

1）AfriNIC（非洲网络信息中心）——非洲区。

2）APNIC（亚太网络信息中心）——亚洲/太平洋区。

3）ARIN（美国因特网地址注册机构）——北美区，包括波多黎各和一些加勒比海岛屿。

4）LACNIC（拉丁美洲和加勒比海因特网地址注册机构）——拉丁美洲和一些加勒比海岛屿。

5）RIPE NCC（Reseaux IP Europeens，欧洲网络资讯中心）——欧洲、中东和中亚。

RIR 系统的目标如下。

1）唯一性。为做到全球因特网路由，每个 IP 地址必须是全球唯一的。

2）汇聚。地址空间的层次化分配，保障在因特网上 IP 流量的正确路由。如果没

⊖ 就像每种看似权威的论断一样，总是存在例外！任意播地址典型地是指派给多台主机的，类似的情况是组播地址是共享的。此处这个论断适用于单播地址。

有汇聚，那么路由表会变得零碎不堪，最终导致因特网内部的众多瓶颈。

3）保护管理。不仅对于 IPv4 空间而且对于 IPv6 空间都一样的是，地址空间需要依据实际使用需求进行分配。

4）注册。IP 地址指派的一个公开可访问注册机制消除了二义性，并会有助于排错。这种注册机制被称为 whois 数据库。如今，存在许多 whois 数据库，不仅由 RIR 运营维护，而且由 LIR/ISP 运营维护（针对它们的相应地址空间运行这种机制）。

5）公平性。非歧视的地址分配，基于真实的地址需求而不是长期的“计划”。

正如您可能猜到的，我们在本章讨论的分配方法类似于如下方法：向 RIR 分配地址块，由 RIR 向国家或本地因特网注册权威机构分配，接下来由这些机构分配到服务提供商和端用户的那些方法。RFC 2050（24）文档的内容是 RIR 地址分配指南。通用地址分配层次结构如图 3-8 所示。国家因特网注册机构与本地因特网注册机构相近，但被组织在国家层次上。

回到 20 世纪 80 年代和 90 年代早期，许多公司（图 3-8 中的端用户）直接从一个中心式因特网网络信息中心（NIC）处得到地址空间。但是，在转换到 CIDR 寻址结构的过程中，为了提供地址分配职责的进一步委派，插入了 RIR 和 LIR/ISP 层。如今，多数机构都从 LIR 或 ISP 处得到地址空间。虽然为了最大化地址效率，建议 RIR 使用一致的地址分配策略，但得到这种地址空间的过程通常由 LIR/ISP 所垄断（dictated），各公司通常与他们有商务关系。

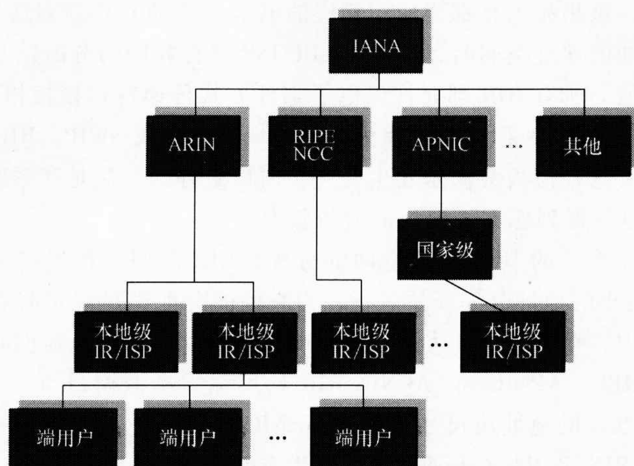


图 3-8 自顶向下的 IP 地址分配[24]

当地址空间被分配给一个 ISP 时，之后该 ISP 可在因特网上通告该地址空间。因此，插入 LIR/ISP 层的这种做法有助于在因特网上汇聚路由通告。由该 ISP 服务的多个顾客可被汇聚在因特网上的一条路由之中（进行通告）。如果商务运转良好，且 LIR/ISP 需要更多的地址空间，那么 LIR/ISP 可从其 RIR 请求更多的空间。一般来说，每个 RIR 有其自己定义的处理地址请求的过程，所以如果要了解细节，您应该咨询您所在地区的 RIR。

3.4.1 RIR 地址分配

从一个 RIR 角度看，RIR 将空间分配给 LIR/ISP，然后 LIR/ISP 将地址空间指派给它们的顾客（客户）。“分配”这个术语，从技术角度看，指将一个 IP 地址块提供作为一个地址空间“池”，可从中抽取地址指派给顾客（客户）。之后像 IPAM 全球公司这样的顾客可使用被指派的地址空间，从该空间分配地址块和子网，接下来从所分配的子网中向各台主机指派 IP 地址。这种分配和指派的过程机制，依据的是我们在本章中前面有关 IPAM 全球公司层次化分配范例中描述过的过程。但是，RIR 将分配和指派作了区分，原因是指派由使用中的地址组成，而分配是用于指派的地址池，它开始时是没有使用的，但理论上是随时间推移从分配中进行多次指派，使用率得以增加。从技术角度看，RIR 将分配和指派都看做是在使用中的，但却将针对实际地址利用率而进行分配空间的审计能力，作为开放空间，将其留作需要时处理来自每个 LIR/ISP 的附加分配请求。

为了在第一时间得到地址空间，LIR/ISP 必须展示说明对直接地址分配的 25% 利用率的需求，以及在 1 年内 50% 利用率的需求[⊖]。这项要求适用于得到 IPv4 空间的过程；如今要得到 IPv6 空间，要比较容易些。如果要请求附加的地址空间，那么就需要通过展示 LIR/ISP 的当前分配的利用率，来说明附加请求的合理性。为了保持跟踪 LIR/ISP 分配，各 RIR 要求每个 LIR/ISP 都实现电子（自动）更新机制。当 LIR/ISP 指派地址空间时，该指派信息都要使用相应的电子（自动）更新表通知 RIR。从理论上说，在请求附加的地址空间时，RIR 和 LIR/ISP 都有共同的分配信息，据此可确认并批准利用率阈值。所有 RIR 都允许以电子邮件方式传递特定模板格式的这种信息，以此传递分配信息。ARIN 称这个过程为共享的域名过程或 SWIP。RIR 会改变电子邮件模板，而且确实这种模板会偶尔发生变化，所以最好的办法是联系服务于您所在地理区域的 RIR，以便得到您需要模板的最新版本。

为了控制正在缩减的 IPv4 地址空间的分配，RIR 采用一种慢启动分配方案，使 LIR/ISP 以一个较小的地址分配开始运行，当它们的分配满足规定时在得到较大的分配。在一些情形中，在 LIR/ISP 从其空间向端用户分配地址时，他们必须从 RIR 获得准许（同意）。RIPE、APNIC 和 LACNIC RIR 利用称为指派窗口（AW）的一种结构来控制需要 RIR 准许的地址块尺寸分配；AfriNIC 使用一种类似的结构，称为亚分配窗口（SAW）。ARIN 采用一个标准的/20 作为有效的 AW。以 CIDR 表示法表示的 AW 或 SAW，加强了边界条件，即在没有从相应 RIR 获取准许条件下，约束 LIR/ISP 从其地址空间分配的地址块最大尺寸。

虽然 APNIC 和 LACNIC 以 AW（和 AfriNIC 的 SAW）作为在没有 RIR 准许条件下的最大分配尺寸，同时 RIPE 使一个 LIR/ISP 至多分配其 AW 的 400% 或至多一个/20，哪个较小分配哪个。因此，拥有一个/22 AW（从 RIPE 得到）的一个 LIR/ISP 至多可

⊖ 注意，随着可用 IPv4 空间的快速消失，分配策略也在快速地演变，并变得日益严苛起来。请联系您的 RIR，了解最新的分配策略。

进行/20 大小的个体地址分配（记住，向左移动 1bit 就会使地址空间加倍），而拥有一个/21 AW 的 LIR/ISP 至多可进行/19 大小的个体地址分配。AfriNIC、RIPE、LACNIC 和 APNIC 将对于新 LIR/ISP 的［S］AW 约束为 0，要求其所有分配都要获得 RIR 准许。随着时间消逝，当 LIR/ISP 证明是一个负责的地址空间管理者时，［S］AW 就可得以增加。表 3-1 汇总了如今 RIR 采用的主要分配策略。

表 3-1 RIR 分配策略汇总

RIR 策略重点	区域因特网注册处				
	AfriNIC(25)	APNIC(26)	ARIN(27)	LACNIC(28)	RIPE NCC(29)
初始阶段的最小 IPv4 分配	/22	/21	/20	/21	/21
IPv4 合理性判定:初步阶段/年	25/50%	25/50%	25/50%	25/50%	25/50%
增加分配的 IPv4 网络利用率准则	80%	80%	80%	80%	80%
对于大于如下情况的分配,必须通知 RIR	指派的 SAW	指派的 AW（最大为/19）	/19(或对于超大的 ISP 为/18)	指派的 AW（最大为/21）	/20 或 4X AW
初始阶段的最小 IPv6 分配	/32	/32	/32	/32	/32
预期的 LIR/ISP IPv6 指派	/48	/48	/48	/48	/48
增加分配的 IPv6 HD 比率准则	0.94	0.94	0.94	0.94	0.94

为了加强 RIR 系统的汇聚目标，当 LIR/ISP 的顾客决定改变服务提供商时，RIR 强烈建议每个 LIR/ISP 以合同方式强制他们的顾客返回分配给他们的地址空间。这个目标的存在是为了保持寻址的层次结构，这类似于我们在 IPAM 全球公司的情形中描述的寻址层次结构。为了形象地说明这项要求的重要性，考虑一个 ISP，它拥有从一个 RIR 得到的一个/20 地址分配。该 ISP 将一个/23 分配给它的一个顾客。如果该顾客改变服务提供商，并将/23 地址空间带走，那么该/23 将需要在因特网“骨干”上汇总（roll up）到新的 ISP，而不是原始的 ISP。原始 ISP 现在将不仅需要通告它最早接收到的/20 地址块，而且还有其他的/23、一个/22 和一个/21 即它所有的地址空间减去离去顾客的/23。新的 ISP 将需要通告它分配得到的地址块，加上新顾客的/23。这明显地在双方的路由表项和全球因特网的路由表项中都产生了增加，挫败了汇聚的目标。这种“可携带的”地址空间被称为提供商无关的（PI），对于客户而言是方便的，但对于因特网路由而言是没有效率的。

采用要求将该地址空间返回原始 ISP 的方法，该/23 就被返回到地址池，可用于在其他地方的指派或进行二次分配，客户将需要以新 ISP 指派的新地址空间对他们的网络重新编址。在改变 ISP 时，要求返还地址空间的做法，确保该 ISP 有单一的汇聚路由通告，被称为提供商汇聚（PA）空间。在分配层次结构中设置 ISP/LIR 层之前

得到 IP 地址空间的一些组织机构，拥有 PI 空间。现在总体而言人们不同意分配 PI 空间，而倾向于 PA 空间。

3.4.2 地址分配效率

在 IPv6 发展的过程中，为得到 128bit 大小的地址，人们进行了许多思考。虽然 IPv4 提供了一个 32bit 的地址字段，它可提供理论上最多 2^{32} 个地址，或 42 亿个以上的地址，在实际中理论最大值要远小于 42 亿个地址。这是由于地址空间的层次化分配导致的，该层次化分配涉及多个网络层次，接下来是子网，最后才是主机。RFC 1715（参考文献［30］）提供了地址指派效率的分析，其中建议使用一个对数尺度用作分配效率的度量，定义为 H 比率：

$$H = \frac{\lg(\text{对象数})}{\text{可用比特数}}$$

如今在因特网上大约有 7.3 亿台主机，则如今的 H 比率 0.277。42 亿个 IP 地址 100% 利用率对应的 H 比率为 0.301[㊀]，所以如今的 H 比率是较高的。

由于 IPv6 巨大的地址空间，IPv6 指派效率的度量并不是依据 H 比率来计算的；其使用一个不同的比率，即 HD 比率[31]：

$$HD = \frac{\lg(\text{被分配对象数})}{\lg(\text{被分配对象最大数量})}$$

在 IPv6 HD 比率中被度量的“对象”是从一个给定尺寸的 IPv6 前缀中指派的 IPv6 站点地址（/48）。/48 地址块是那些由 LIR/ISP 向每个端用户期望指派的地址块。所以拥有一个/32 地址分配的 LIR/ISP，如果已经分配 100 个/48 地址块，那么它将具有的 HD 比率是 lg（100）/lg(65536) = 0.415。

3.5 多穴接入法和 IP 地址空间

“多穴接入法”这个术语指一个企业拥有到因特网的多条（>1）连接。一个简单的架构如图 3-9 所示。多穴接入战术策略提供几项优势。

1）链路冗余，在出现连接中断时可提供连续的因特网连接可用能力。

2）ISP 冗余，在出现一个 ISP 中断时，多个 ISP 被用来限制被影响面（exposure）。

3）在多条连接上进行因特网流量的负载均衡。

4）策略和性能优势，可通过基于拥塞的流量路由或基于需求而将不同应用的流量路由到不同的链路或 ISP 获得这些优势。

虽然在将多穴接入法配置到每个 ISP 的路由器接口时，要求特别谨慎，但它确实提供了几项有吸引力的优势。如我们在图 3-9 所示的，企业边界路由器直接与它们相应的 ISP 边界路由器对接，参与到一个外部路由协议（例如 BGP）之中，通告到相应

㊀ 0.301，它等于 lg（2），是 H 比率的最大值。

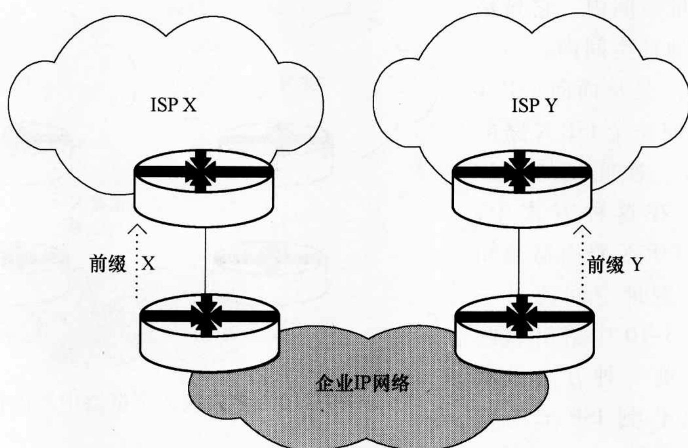


图 3-9　多穴接入法的架构

地址块的可达性（通过地址前缀）。因此，连接到 ISP X 的企业路由器，将通告由 ISP X 提供给该企业的地址空间的可达性。类似地，连接到 ISP Y 的企业路由器，将通告由 ISP Y 提供给该企业的地址空间的可达性。

这两台企业路由器也通过企业 IP 网络使用一个内部路由协议，相互通信。采取这种方式，可检测到一个 ISP 连接的丢失，虽然就这点而言也是引起人们关注的地方。为了形象地说明这点，而不涉及所有的路由细节条件下[⊖]，下面汇总了最常见的多穴部署选项、中断影响以及对 IP 地址空间意味着什么。

1）情形 1。两条或多条不同的物理链路连接到同一 ISP。这种“多连接”(multiattached) 结构提供了链路冗余，但没有提供 ISP 冗余。图 3-9 所示，将两个 ISP 云缩为单一云，但仍然有来自该企业的两条（或更多）链路。采用一个 ISP 时，前缀 X = 前缀 Y，所以从该 ISP 分配的这个公开地址空间可被统一地在所有连接上进行通告。

2）情形 2。到一个或多个 ISP（使用提供商独立的地址空间）的两条或多条连接。回顾一下，在这个场景中，PI 空间被直接分配给组织机构，且独立于 ISP 联盟。图 3-9 所示，被通告的前缀在两条连接上也是相同的，虽然可将其表示为前缀 Z，以便表示独立于 ISP 地址空间。和在情形 1 中一样，PI 空间可被通告到所有 ISP，并依据需要在该机构内实施分配。

3）情形 3。到两个或多个 ISP 的两条或多条连接（使用来自每个 ISP 的提供商汇聚地址空间）。在这种情形中，每个 ISP 将分配地址作为其服务的组成部分。图 3-9 反映了这样的场景。拥有两个独立的地址块 X 和 Y 时，如果到 ISP X 的链路失效，则通过来自连接到 ISP X 的企业路由器的内部路由协议更新，连接到 ISP Y 的企业路由器将检测到链路失效。因此，现在连接到 ISP Y 的企业路由器可通告到前缀 X 的可达性。取决于 ISP Y 的策略，它也许会传播也许不会传播该路由，原因是该路由没有落

⊖　请参考如下 RFC 了解多穴接入法的路由细节：2260[170]、4116[171]、4177[172]和 4218[173]

在 ISP Y 的地址空间内，它只是落在 ISP X 的地址空间内。

另一种方法是从面向 ISP Y 的企业路由器向一个 ISP X 路由器对等端执行一次间接的 BGP 对等端更新。在这种方式中，通过 ISP Y 向 ISP X 路由器通知到达企业所属地址空间的另一条路由。在图 3-10 中给出这两种替代方法，前一种方法显示为前缀 X 被通告到 ISP Y 的路由器，后一种方法被显示为被通告到 ISP X 的路由器。

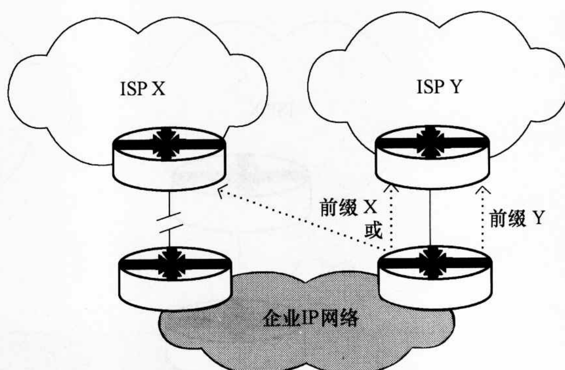


图 3-10 多穴接入下链路中断的恢复

在每条 ISP 连接处的 NAT 网关通常部署时都激活带有地址池能力，依据 ISP 连接的邻近性（例如邻近前缀 X 或前缀 Y）将一个给定报文的内部私有地址转换为一个公开地址。通过禁止使用 NAT 网关，企业边界路由器策略应该被实现为：在内部寻址的主机间最小化或防止经过该 ISP（可能是多个 ISP）而路由流量。

3.6 地址块分配和 IP 地址管理

在本章，我们讨论了公开和私有 IPv4 空间和 IPv6 空间的 IP 地址块分配技术。从基本管理角度来看，就分配的层次结构方面维护地址分配的一个清单是至关重要的，在这个清单中，部署了子网的位置，还有任何其他相关信息，例如每个子网到其对应服务路由器或交换机接口的映射、在合适情况下会有关联的因特网注册机构的管理信息、问题（trouble）联系信息以及人们关注的其他信息。

负责地址块分配的许多组织机构，使用表格或简单的数据库应用（例如微软 Access）保持这种信息处于有结构状态。虽然执行地址块分配和二进制算术运算，不能轻松地通过单一鼠标点击来操作表格，但低层次本机运行（underlying homegrown）的 Visual Basic 或 perl 代码，可应用在本章讨论过的逻辑，目的是实施最佳拟合、稀疏或随机地址分配，并跟踪记录得到的分配。当然，要准确地实施这些操作，并管理地址块分配、移动、修改和删除，必须小心谨慎为好。

保持 IPAM 全球公司地址分配表格最新的方法是一有分配就更新该表格，从而可反映我们的私有 IPv4、公开 IPv4 和 IPv6 以及 ULA IPv6 地址分配。这种信息为下一步操作（从相应子网为每个位置分配独立的 IP 地址）提供了必要的基础。既然每个分支办事处和配送中心都有 IP 子网可向上汇聚到该层次结构，那么就可开始地址指派过程了。我们将在本书下一部分中描述自动化地址指派的一种方式，即 DHCP，我们将在第 14 章中讨论地址指派的其他多种手工配置方法。

第Ⅱ部分　动态主机配置协议（DHCP）

第Ⅱ部分讲述了 IPv4 和 IPv6 的动态主机配置协议（DHCP）。在技术综述之后，讲述了由 DHCP 支持的各项应用，最后讲述其部署和安全考虑因素。

第 4 章　DHCP

4.1　引言[㊀]

在因特网存在的早期日子里，当主机数处于数百台量级时，将一个 IP 地址指派到一台设备是相当简单平凡的。简单地说，即在每台主机上手工输入配置参数之一。这种使用一个硬编码 IP 地址的“一次即可”或静态地址指派过程，当然是简单的，但它却约束了在不同网络或子网间的主机移动性。支持移动性的做法，要求基于当前位置或网络（预期主机将与其连接）以一个新 IP 地址对主机重新配置的烦琐任务。

尽管如此，您将可能有一组静态地址用于您所在机构网络上不需要移动性的设备，例如路由器、服务器、IP PBX 等。在被分配子网上哪些 IP 地址是静态指派的、哪些 IP 地址指派到地址池用于动态指派以及哪些 IP 地址是空闲的或预留作为未来使用，保持跟踪记录这些信息是必要的。一项建议的实践措施是维护一个子网 IP 清单，维护在每个子网上被指派的地址记录，以便最小化重复 IP 地址指派或产生错误的 IP 地址指派问题。仅需要确保静态地址清单要与实际配置在路由器、服务器或静态寻址设备上的地址相匹配，就可以了。通过各种形式的发现或 ping 大片扫描（sweep）手段，实施地址指派的周期性基线检查，可帮助识别任何地址指派的不匹配问题，这将在第 14 章中讨论。

4.2　DHCP 综述

动态主机配置协议（DHCP）是一个客户端-服务器协议，被连接到一个 IP 网络的设备用来自动地获得一个 IP 地址。DHCP 是能为 IP 网络管理员节省大量时间的装置。它使一台设备能够广播对一个 IP 地址的请求，在没有用户介入的条件下，在该 IP 网络内部有一台或多台 DHCP 服务器服务这条请求。对于多数端用户设备（例如

㊀　本章开始各节的依据是参考文献［11］的第 3 章。

笔记本计算机、VoIP 电话、PDA 和其他设备）而言，在设备启动或连接到一条有线网络或无线网络时，在没有用户介入的条件下，DHCP 过程在“后台”运行并发挥作用（transpire）。DHCP 也支持 IP 地址的高效使用，方法是允许在动态分配的地址池内，在设备间重用一个 IP 地址。一个给定的 IP 地址在一天为一台设备所用，在第二天可以被另一台不同的设备所用。

DHCP 是作为 IPv4 和 IPv6 的组成部分得以支持的。我们将在本章讨论（DHCP 的）IPv4 版本，在下一章讨论（DHCP 的）IPv6 版本。当我们在本章中使用“DHCP”这个术语时，我们指的是 DHCP 的 IPv4 版本。DHCP 是在 RFC 2131[32] 和 2132[33] 中定义的，且在后续多个 RFC㊀增加了许多内容，DHCP 是构建在一个较老的协议启动协议（被称作 BOOTP）的基础上的。BOOTP 最初是在 RFC 951[34] 中规范下来的，提供了地址指派的自动化方法，但受限于将一个给定 IP 地址预指派到一个特定设备的做法，其中以该设备的网络接口（MAC）地址识别它的。因此，一台 BOOTP 服务器配置有一个 MAC 地址列表以及对应的 IP 地址。在不要求每个客户端的硬件地址的先验知识条件下，DHCP 将这项功能与将 IP 地址指派到客户端的增值能力集成在一起。实际上，DHCP 取代了 BOOTP，支持与 BOOTP 客户端的后向兼容（互操作）。

DHCP 支持三种类型的 IP 地址分配。

1）自动化的地址分配。DHCP 服务器将一个永久的 IP 地址指派给客户端。

2）手工地址分配。像 BOOTP 一样，DHCP 服务器基于特定设备的硬件地址，指派一个“固定的”IP 地址。

3）动态地址分配。DHCP 服务器指派一个 IP 地址是有限时间段的，在该段时间之后，该地址可被重新指派，有可能被指派到另一台不同的设备。

对于要求一个永久 IP 地址指派的特定用户和设备集合而言，通过 DHCP 的自动地址分配是有用的，在这种情形中，对于一个特定用户或设备是否要有一个特定 IP 地址，是没有要求的。换句话说，你可预留许多“永久的”地址，这些地址没有直接将每个 IP 地址与一个特定的硬件地址关联起来。这种做法是与手工 DHCP 形成鲜明对比的，后者将一个特定的硬件地址与一个对应的 IP 地址关联起来。

为了“重用”IP 地址，通常情况下，动态地址分配在 DHCP 服务器中设置地址池。在动态分配情况下，DHCP 服务器将其 IP 地址租给客户端一个固定的时间段。采用这种方式，DHCP 服务器在一个给定时间段（称作租期时间）内将一个 IP 地址指派给一个特定的客户端，当由于租赁过期或客户端放弃地址时，这时该 IP 地址就是可用的了，DHCP 服务器就会将该地址重新指派给一个不同的客户端。在 DHCP 服务器实现内部，租赁时间是一个可配置的参数。

不管 DHCP 地址分配类型为何，一个 DHCP 客户端获取一个地址租赁的过程都是相同的。基本过程开始时，都是一个 DHCP 客户端广播一条 DHCPDISCOVER 报文。因为客户端既没有一个 IP 地址，通常也没有有关 IP 网络的任何信息，所以它将全零

㊀ 要得到一个完整列表，请参见本书后面的 RFC 索引。

地址作为源地址、将广播（全 1）地址作为目的地址插入到 IP 首部内部。让我们假定一台 DHCP 服务器部署在 DHCP 客户端所连接的相同子网上。在接收到 DHCPDISCOVER 报文时，DHCP 服务器将确定在 DHCPDISCOVER 接收到的这个子网上，它是否有一个地址可用。

如果在地址池中有一个地址可用，那么该 DHCP 服务器将向客户端发送一条 DHCPOFFER，提供一个 IP 地址和关联的配置参数（称为选项（options））。如果有多台 DHCP 服务器正在服务这个子网，那么 DHCP 客户端可能接收到一条以上的 DHCPOFFER。客户端将选择一个配置集合，并广播一条 DHCPREQUEST 报文，确定被选中的 DHCP 服务器（客户端从之接收到提供的 IP 等参数）。一旦被选中的 DHCP 服务器将地址租赁信息记录在非易失存储之中，它将以一条 DHCPACK 确认该条 DHCPREQUEST，从而将该 IP 地址绑定到 DHCP 客户端。这个基本的消息流，如图 4-1 所示，有时被称为“DORA”过程——发现（Discover）、提供（Offer）、请求（Request）和确认（Ack）。

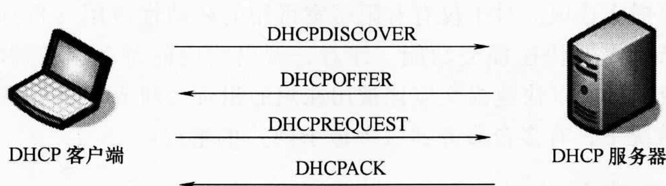


图 4-1　DHCP“DORA”过程

在这个简单的范例中，DHCP 服务器驻留在与 DHCP 客户端相同的子网上。客户端在该网络上广播 DHCPDISCOVER 报文。因为 DHCP 服务器驻留在相同的网络上，那么它接收到广播报文，并对之进行处理。因为知道广播所发出的网络（信息），所以该 DHCP 服务器就可指派在该网络上一个可用的 IP 地址。但您是否必须在每个子网上都要部署一台 DHCP 服务器呢？幸运的是，不需要；简单地说，通过 IP 路由基础设施，从子网必须可到达该 DHCP 服务器即可。接收到 DHCPDISCOVER 广播报文的路由器（可能是多台）将不会传播广播报文（新报文），因为这将造成过度的和不必要的 IP 流量。相反，该路由器将转发或中继这条广播报文到目的 DHCP 服务器（可能是多台）。被配置执行这项中继功能的每台路由器被称作一个中继代理。每个中继代理必须以服务该子网的每台 DHCP 服务器的 IP 地址进行配置。这个配置参数，常被称作 DHCP 中继地址，使路由器能够接收 DHCPDISCOVER 广播报文，查找被配置成 DHCP 中继的 DHCP 服务器（可能是多台），之后将 DHCPDISCOVER 报文通过单播直接传输到每台 DHCP 服务器，如图 4-2 所示。

在上述过程中，该路由器修改 DHCPDISCOVER 报文，将接收到 DHCPDISCOVER 的那个接口的 IP 地址插入到中继代理（网关）接口地址字段。这个参数使 DHCP 服务器能够识别请求一个地址指派的那个子网。注意，当网关接口地址（GIAddr）字段为零时，DHCP 服务器会假定要被指派 IP 地址的子网与接收到 DHCPDISCOVER 的那个子网（通过直接广播接收到报文）是相同的。

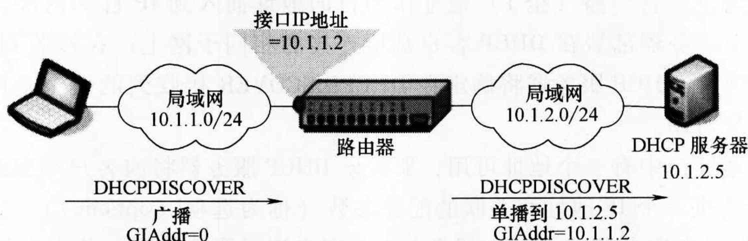


图 4-2 DHCP 中继[11]

除了上面列出的四条报文的交换过程外，IETF 采用了 RFC 4039，该 RFC 定义了一个快速提交选项，即选项 80。这个选项是在下一章定义的 DHCPv6 相应内容之后才形成的，它将消息传播需求分成两部分，方法是作为对一条 DHCPDISCOVER 消息的响应，使路由器简单地发送一条 DHCPACK。客户端将在其 DHCPDISCOVER 消息中包含快速提交选项。以一个地址指派做出响应的各服务器将直接发出一条 ACK 报文，它也包含该快速提交选项。对于仅有有限带宽可用的移动性应用（例如蜂窝（移动）电话）而言，特别需要快速提交功能。注意，做出响应的每台服务器将假定它所指派地址是被租赁的，所以快速提交应该被用在短的租赁时间或仅由有限数量服务器支持（如果正常情况下，有多台服务器服务该子网）的情况。

4.2.1 DHCP 消息类型

我们介绍了四种基本 DHCP 消息类型，所以让我们在此基础上进行扩展，并浏览一下 DHCP 消息的全集及其相应含义。通常我们忽略了这些消息上的“DHCP”前缀，且仅将第一个字母大写，但下面是它们正式被定义的情况。

1）DHCPDISCOVER。从客户端向服务器发出的，用于请求 DHCP 地址指派；DH-CPDISCOVER 会包括该客户端要求的参数或选项。

2）DHCPOFFER。作为对一条 DHCPDISCOVER 的响应，是从服务器向客户端发出的，向客户端指明这是一个 IP 地址提供，其中包括它对应的租赁时间（和其他配置参数）。

3）DHCPREQUEST。作为对一条 DHCPOFFER 的响应，是从客户端向一台服务器发出的，用于接受或拒绝被提供的 IP 地址，其中还包含期望的参数设置或其他参数设置。期望延长或重新请求其现有 IP 地址租赁的客户端也可使用 DHCPREQUEST。

4）DHCPACK。是从服务器向客户端发出的，用于正面确认 IP 地址租赁和关联参数设置的授权。从现在开始，客户端开始使用该 IP 地址和参数数值。

5）DHCPNAK。是从服务器向客户端发出的，用于负面地确认 DHCP 事务。该客户端必须释放使用该 IP 地址，如有必要，它需要重新发起该过程。

6）DHCPDECLINE。是从客户端向服务器发出的，用于指明由服务器提供的这个 IP 地址已经被另一个客户端所用。那么，典型的操作是，DHCP 服务器将该 IP 地址标记为不可用的。

7）DHCPRELEASE。是从客户端向服务器发出的，将该客户端正在释放 IP 地址

的信息通知服务器。之后，客户端必须释放该 IP 地址的使用。

8）DHCPINFORM。是从客户端向服务器发出的，用于从服务器请求非 IP 地址配置参数。服务器将形成一个 DHCPACK 应答，带有合适的相关联数值。

9）DHCPFORCERENEW。是从服务器向客户端发出的，为了使一个客户端得到一个新（不同）的 IP 地址，强制该客户端进入 INIT 状态[⊖]。几乎没有哪种客户端实现了对这条消息的支持。

10）DHCPLEASEQUERY。是从一个中继代理或其他设备向服务器发出的，用于向 DHCP 服务器询问一个给定的 MAC 地址、IP 地址或客户端-标识符等数值是否有一个活跃的租赁及关联的租赁参数数值（主要由宽带接入集线器或边缘设备使用）。

11）DHCPREALSEUNASSIGNED。是从服务器向一个中继代理发出的，作为对一条 DHCPREASEQUERY 的响应，通知该中继代理，指明这台服务器支持那个地址，但却没有活跃的租赁在用。

12）DHCPREASEUNKONWN。是从服务器向一个中继代理发出的，作为对一条 DHCPREALSEQUERY 的响应，通知该中继代理，指明这台服务器不知道在请求中所指定的客户端。

13）DHCPREASEACTIVE。是从服务器向一个中继代理发出的，作为对一条 DHCPREALSEQUERY 的响应，其中带有端点位置和剩余的租赁时间。

RFC 2131 定义了一些状态，就使用 DHCP 的 IP 地址配置方面，客户端存在于这些状态之中。定义了如下状态。

1）INIT。初始化，意味着该客户端既没有一个 IP 地址也没有任何以前的配置信息。

2）INIT-REBOOT。虽然客户端有以前的 IP 地址信息，但它开始初始化，期望确认其配置。

3）BOUND。客户端和服务器同意它们之间的 IP 租赁协议。

4）RENEWING。客户端正在尝试刷新地址租赁。

5）REBINDING。客户端的租赁超期时间正在逼近，它正在尝试刷新地址租赁。

6）SELECTING。中间状态，其中客户端正在等待并评估来自 DHCP 服务器（可能多台）的多条 DHCPOFFER。

7）REQUESTING。中间状态，其中客户端已选择一条提供（Offer），并期望接收该条提供，或已经识别一条提供，指明一个 IP 地址已在使用，在这种情形中它向服务器发送一条 DHCPDECLINE。

8）REBOOTING。在一次重启之后，客户端正在尝试重新绑定。

简言之，刷新和重新绑定之间的区别在于刷新请求的紧急程度和传输模式，其中重新绑定具有较高的紧急程度，刷新是单播的，重新绑定是广播的。当初步得到一个地址租赁时，客户端设置两个定时器：

1）T1 = 50% 的租赁时间，这是默认的数值。

⊖　接下来我们将讨论 DHCP 状态。

2）T2 = 87.5% 的租赁时间，这是默认的数值。

通过在 DHCP 报文交换内部在相应的选项内指定的数值，依据 DHCP 客户端和服务器之间的一致协议，可修改这些定时器数值。在 T1 定时器超期时，客户端进入刷新状态，并尝试刷新地址租赁，方法是向它得到地址租赁的 DHCP 服务器，单播一条 DHCPRENEW 消息。如果接收到一条 DHCPACK，则客户端重新进入绑定状态。如果接收到一条 DHCPNAK，则客户端释放该 IP 地址的使用，并进入 INIT 状态。在其他情况下，即没有接收到一个响应，那么客户端等待 T2 定时器的超期，之后广播一条 DHCPRENEW，尝试刷新地址租赁。在原始服务器（地址租赁是从这里得到的）下线且存在一台故障切换 DHCP 服务器可刷新地址租赁时，发出广播请求。

图 4-3 给出这些 DHCP 客户端状态间的状态转换图和对应的状态转换机制。注意图中没有包括 DHCPINFORM 消息，原因是这条消息与非 IP 地址参数有关。已配置一个 IP 地址的客户端发出 DHCPINFORM 消息，请求其他参数设置，服务器以一条 DHCPACK 应答，指明被请求的值。

图 4-3 DHCP 状态转换[32]

4.2.2 DHCP 报文格式

让我们仔细研究 DHCP 报文⊖中的各字段以及这些字段是如何与整个 DHCP 过程相关的。图 4-4 显示字段布局，我们将在该图后面描述每个字段。

DHCP 报文字段描述：

（1）操作码。利用 BootP 这个前辈协议，这个字段的值是

1）1 = BootRequest

2）2 = BootReply

⊖ 一条“DHCP 报文”是在一条 IP 报文内部的 IP 之上传输的。DHCP“应用”使用在本章中讨论的报文首部格式。

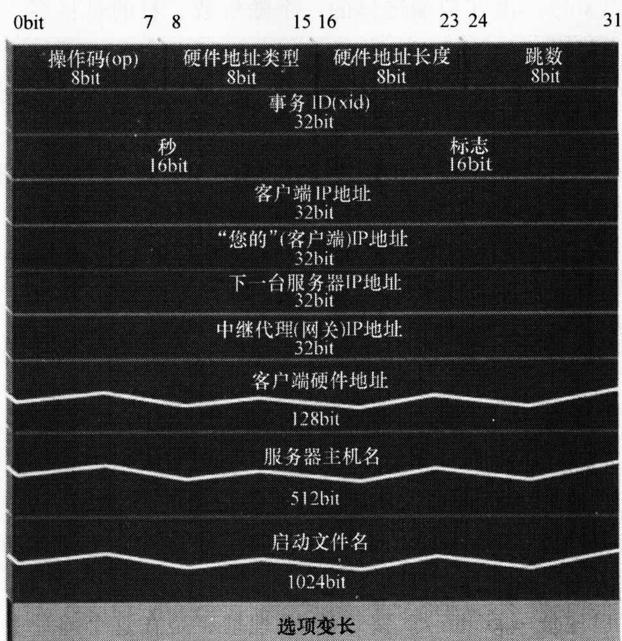


图 4-4 DHCP 报文的各字段[32]

注意 DHCP 消息类型（Discover（发现）、Offer（提供）、Request（请求）等）实际上是在选项字段中采用选项号码 53（DHCP 消息类型）定义的，具有如下有效数值。

1）1 = DHCPDISCOVER

2）2 = DHCPOFFER

3）3 = DHCPREQUEST

4）4 = DHCPDECLINE

5）5 = DHCPACK

6）6 = DHCPNAK

7）7 = DHCPRELEASE

8）8 = DHCPINFORM

9）9 = DHCPFORCERENEW

10）10 = DHCPLEASEQUERY

11）11 = DHCPLEASEUNASSGNED

12）12 = DHCPLEASEUNKNOWN

13）13 = DHCPLEASEACTIVE

（2）硬件地址类型。硬件或 MAC 地址的类型，例如以太网、802 等。

（3）硬件地址长度。定义了以字节为单位的 MAC 地址长度。

（4）跳数。由客户端设置为零，这个字段可由客户端和服务器之间的每台路由器做加 1 处理。

(5) 事务 ID (xid)。由客户端选择的一个随机数，目的是将客户端与服务器之间的消息和响应关联起来。

(6) 秒 (secs)。自客户端开始获取一个 IP 地址或刷新的过程以来，消逝的秒数。

(7) 标志。这个字段由这样的 DHCP 客户端使用，直到它的 IP 协议软件被配置之前，它是不能接受单播 IP 报文的。对于这样的情形，客户端将这个字段中的第一个比特设置为 1，并将剩余比特都设置为 0。当设置为 1 时，服务器（如果是局域网方式连接的）或中继代理将向客户端广播 Offer 和 Ack 消息；否则，服务器或中继代理将这些消息发送到 yiaddr 字段中指定的单播地址。这个比特有时被称为标志字段中的广播比特。

(8) 客户端 IP 地址 (ciaddr)。客户端使用的 IP 地址，这是当客户端知道该地址时才使用的，例如处于 BOUND、RENEWING 或 REBINDING 状态时的情况。

(9) 提供的 IP 地址 (yiaddr)。DHCP 服务器指派的 IP 地址，将由客户端使用。

(10) 服务器 IP 地址 (siaddr)。用于启动的“下一个”服务器的 IP 地址，是由 DHCP 服务器提供的。

(11) 网关接口地址 (giaddr)。接口的 IP 地址，是在这个地址上接收到 DHCP 广播的，它由中继代理发出。

(12) 客户端硬件地址 (chaddr)。客户端提供的客户端的链路层或硬件地址。

(13) 服务器名 (sname)。DHCP 服务器主机名。

(14) 文件。启动文件名，为 null（空）或完全符合格式的目录路径名。

(15) 选项。其他 IP 参数，例如租赁时间、域名、缺省网关和子网掩码（要了解一个完整列表，请见下一节）。选项字段的首个四字节总是魔术 cookie 数值（以十六进制表示）：63825363。这是从 RFC 951 的原始 BootP 规范中继承下来的，目的是出于特定厂商的目的（比如）来解释选项的一种方式。

4.3 DHCP 服务器和地址指派

每台 DHCP 服务器可被配置带有多个地址池，这些地址池服务于许多位置的数个不同子网。事实上，对于一些 DHCP 服务器实现来说，出于冗余性考虑，可在多台 DHCP 服务器上配置相同的地址池。在第 7 章将比较详细地讨论这点内容。在 DHCP 服务器配置地址池的所有地址中，它跟踪所有 IP 地址的状态。当一个地址租赁给一个客户端时，一般情况下，该服务器跟踪的不仅是该 IP 地址的租赁时间，而且还包括租赁该 IP 地址的客户端的一个标识符。虽然也可使用客户端标识符字段，即选项 60，但典型情况下，这个标识符是客户端的 2 层 (MAC) 地址，是通过 chaddr 字段得到的。

建议在 chaddr 字段上使用客户端标识符（客户端 ID），目的是维护该设备的一个标识符，即使在链路硬件被拆除并安装到另一台设备上时也要使用该标识符。但在实际中，多数设备没有提供一个客户端 ID，或将 chaddr 字段的值复制到客户端 ID

选项。

典型情况下，在提供一个地址时，DHCP 服务器使用的基本决策过程依据如下信息：

1）如果客户端有一个租赁的地址（记录在 DHCP 服务器中），则该服务器将指派这个地址。

2）如果客户端以前有这样一个地址，现在过期了或被释放了，但仍然还是可用的，则该服务器将指派这个地址。

3）如果客户端在被请求 IP 地址选项（选项 50）中包含一个地址，且该地址是可用的，则服务器将指派这个地址。

4）服务器将从满足如下条件的子网的一个地址池中指派一个可用的地址，即如果 GIAddr 字段为零，则该子网为接收到 DHCPDISCOVER 广播的子网，如果 GIAddr 值非零，则该子网为由 GIAddr 指明的子网。另一个准则依据 DHCPDISCOVER 报文内部各参数，如果有多个地址池服务存在疑问（不知道使用哪个地址池时）的子网，该准则可决定从哪个地址池中得到指派。一般地说，这些参数被称作客户端类（class）参数，接下来将讨论该内容。

4.3.1 依据类的设备识别

客户端类参数为 DHCP 客户端向 DHCP 服务器提供附加信息提供了一种方法，同时也为 DHCP 服务器识别要求唯一 IP 地址或参数指派的客户端提供了一种方法。例如，您也许希望为 VoIP 设备专用一个地址池，为数据设备专用一个独立的地址池。这种做法的动机来源可能是管理问题或源路由策略（来自相应设备的语音报文与数据报文的不同策略）。多数 DHCP 服务器（包括可从因特网系统联盟（ISC）和微软公司得到的那些 DHCP 服务器）支持将匹配的厂商类或用户类的数值加以指定的作法，这样做的目的是提供这样的分类处理。在从一个地址池中指派一个地址时，依据准则，DHCP 服务器可被配置成关联厂商类或用户类的一个特定数值或一组数值。

让我们考虑一个范例。回顾第 3 章中为 IPAM 全球公司分配地址空间时，我们为旧金山的 VoIP 设备分配子网 10.16.128.0/23。许多组织机构在一个位置分配单一子网，由于不同设备在初始化和配置要求方面的不同，它为不同的 VoIP 设备厂商定义两个独立的地址池。在 IPAM 全球公司的情形中，我们将在 10.16.128.0/23 子网内部为“厂商 X”的 VoIP 设备定义一个地址池，在同一子网内为“厂商 Y”的 VoIP 设备定义一个不同的地址池。那么我们在 IPAM 全球公司的 DHCP 服务器上为每个地址池定义一个池（共两个地址池），比如 10.16.128.20 ~ 10.16.128.250 地址区间和 10.16.129.20 ~ 10.16.129.250 地址区间。出于简单性考虑，我们将它们表示为相等的尺寸大小，但并不要求这样做。在我们的 IP 子网清单中，无论在一个表格、数据库或 IP 地址管理系统哪一种表示法中，我们都能够在这些对应的子网内记录这些地址池。我们也应该记录了静态地址指派，例如 10.16.128.1 为一台路由器所用，10.16.128.6 为一台服务器所用等。

在图 4-5a 中，我们希望配置我们的 DHCP 服务器，以便在 VoIP 电话厂商之间做

出区分，并从不同地址池指派地址。第一步是确定在 DHCP 报文中的什么信息可被用于唯一地识别每种设备类（如 VoIP 电话或笔记本计算机）。典型情况下，您的 *VoIP* 电话提供商将通知您，在厂商类标识符选项（选项 60）中有一个特定的字符串；让我们假设，虽然原始但足够可用的假定这个字符串是“vendorX”（用于识别厂商 X）和“vendorY”（用于识别厂商 Y）。

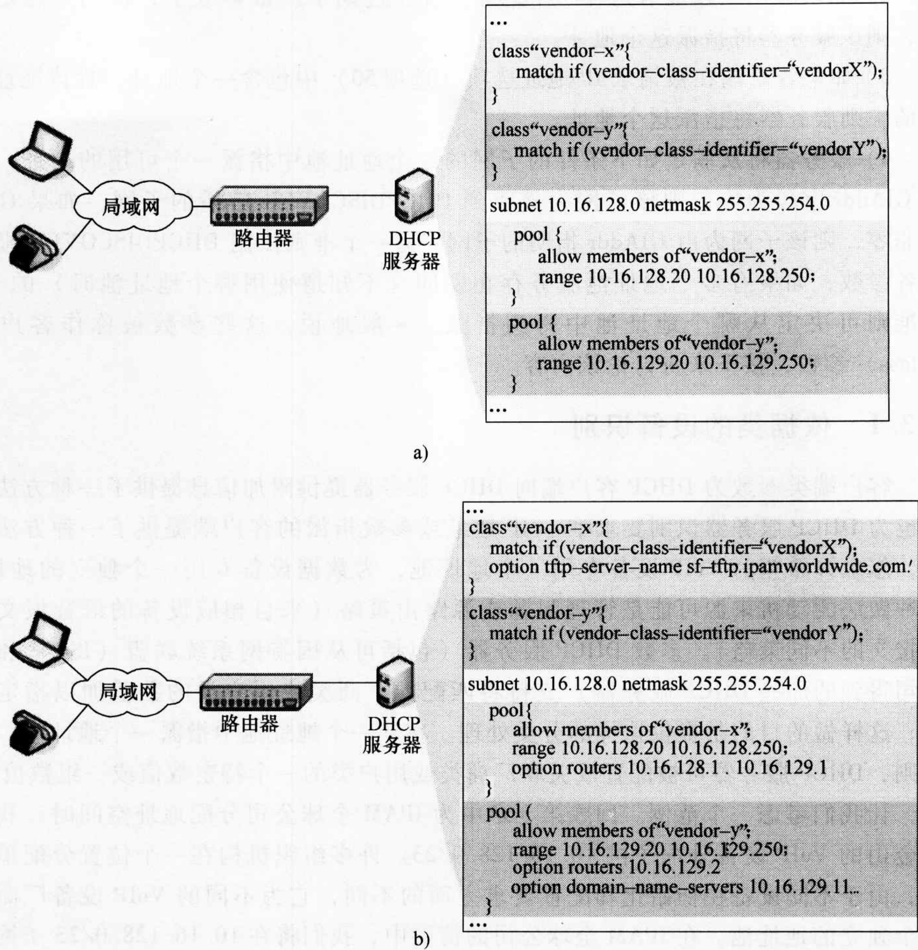


图 4-5 使用 DHCP 配置客户端

a）使用 DHCP 配置伪码的客户端分类范例（依据参考文献［35］）

b）为 DHCP 客户端依据类指定附加的配置信息（依据参考文献［35］）

我们可依据图 4-5a 中的范例，在 DHCP 服务器中为每个厂商定义一个类，但语法将取决于您的 DHCP 服务器厂商（或 IPAM 工具）。在这个范例中，我们配置 DHCP 服务器将发送 DHCP 报文（带有选项 60 = “vendorX”）的设备分类为 vendor-x 类的设备。类似地，通过定义厂商类型标识符选项，依据图 4-5a 中的 match-if（匹配条件）语句，使其具有值“vendorY”，则我们定义了区分厂商 Y 设备的一个类。另外，

可设置第三种地址池，作为不匹配其他已定义客户端类的各客户端的“默认”地址池。

既然我们定义了我们的两个类，这使 DHCP 服务器能够将从属于一个类或另一个类的设备发出的报文加以识别，则现在我们能够命令服务器如何处理这些请求。在对应的子网声明语句内，我们可定义两个地址池，原因是我们希望区分这两个设备类的地址指派。在我们的 10.16.128.0/23 子网内部，我们将 vendor-x 类设备的一个地址池定义为包含地址 10.16.128.20 ~ 10.16.128.250，将 vendor-y 类设备之子网上的第二个地址池定义为包含地址 10.16.129.20 ~ 10.16.129.250，在图中映射到类的部分加了阴影。

当依据图 4-5a 进行配置时，为了区分设备类，现在 DHCP 服务器将检查来自 10.16.128.0/23 子网上设备的每条 DHCPDISCOVER 报文，之后为厂商 X 设备指派来自 10.16.128.20 ~ 10.16.128.250 地址池的一个地址，为厂商 Y 设备指派来自 10.16.129.20 ~ 10.16.129.250 地址池的一个地址。注意，可能存在您希望在每条这样的地址池声明语句内定义的其他参数或选项设置，以便依据每个类的设备提供配置信息，稍后我们将讨论。

取决于您部署的 DHCP 服务器的厂家，会存在多种菜单界面或文本文件编辑器和准则，以这些工具来确定地址指派逻辑。例如，微软的 DHCP 服务器可通过一个 Windows 图形用户界面（GUI）进行配置，而 ISC DHCP 服务器可通过文本编辑器进行配置。但是，除了用户类和厂商类外，在定义客户端类方面，ISC DHCP 提供了更多的灵活性；在进行客户端类处理时，可检查并过滤报文中的任何参数，包括用于 MAC 地址过滤的 chaddr 字段或存在的任何其他参数。对于混合 ISC 和微软的环境，使用一个中心化的 IPAM 系统可有助于抽象各厂商的独特界面，并以单一界面支持两者的配置。

4.4 DHCP 选项

客户端可请求特定选项的设置，而服务器可依据 DHCP 服务器配置指派这些选项和其他参数。DHCP 管理员可定义要向所有或某些 DHCP 客户端指派的选项组，依据的是客户端的硬件地址、客户端类值或其他 DHCP 报文参数。

如在前一节讨论的情况，我们为 IPAM 全球公司的旧金山办事处依据厂商对应的 VoIP 设备，设置两个客户端类。这些类的设备将可能要求不同的配置参数。例如，Cisco VoIP 设备典型地要求选项码 66 或 150，而 Avaya VoIP 设备要求选项 172。我们已经描述过客户端类如何被用来配置 DHCP 服务器，使其区分不同的 DHCP 客户端。现在我们为每个地址池关联选项（可能是多个选项），这些地址池将提供给各客户端，它们从相应地址池中接收地址。这种做法的一个范例如图 4-5b 中高层样例配置，它包括带有类和地址池语句的选项声明，来定义向客户端提供的其他参数。另外，手工 DHCP 地址预留的做法，支持将一个硬件地址映射到一个特定的 IP 地址，也可为设备定义关联的 DHCP 选项。

表 4-1 列出已定义的当前 DHCP 选项集合。“代码”列指明选项码或号，“名称”列给出对应的选项名称。注意“Len”（长度）列指明在选项内部长度字段的数值。选项总长度是这个数值加上两个字节，其中一个字节是代码，一个字节是长度字段自身。

表 4-1 DHCP 选项集合

代码	名称	Len（长度）	含 义	参考文献
0	填充	0	无	RFC 2132[33]
1	子网掩码	4	“IP 地址”格式中的子网掩码	RFC 2132[33]
2	时间偏移	4	以 s 描述的、与 UTC 的时间偏移（RFC 4833 使该值废弃不用，该 RFC 中规范了使用选项 100 和 101）	RFC 2132[33]
3	路由器	*N*	*N*/4① 路由器（默认网关）地址	RFC 2132[33]
4	时间服务器	*N*	*N*/4 时间服务器地址	RFC 2132[33]
5	名称服务器	*N*	*N*/4 IEN-116② 名称服务器地址	RFC 2132[33]
6	域名服务器	*N*	*N*/4 DNS 服务器地址	RFC 2132[33]
7	日志服务器	*N*	*N*/4 MIT 计算机科学实验室（LCS）UDP 日志服务器地址	RFC 2132[33]
8	配额（quotes）服务器	*N*	*N*/4 “当天配额”服务器地址	RFC 2132[33]
9	LPR 服务器	*N*	*N*/4 线式打印机服务器地址	RFC 2132[33]
10	影像（impress）服务器	*N*	*N*/4 图像式影像服务器地址	RFC 2132[33]
11	RLP 服务器	*N*	*N*/4 资源定位服务器地址	RFC 2132[33]
12	主机名	*N*	客户端主机名字符串	RFC 2132[33]
13	启动文件尺寸	2	启动文件的尺寸，以 512 字节块为单位表示	RFC 2132[33]
14	法律依据导出（Merit dump）文件	*N*	在客户端崩溃时，客户端应该将其内核映象导出到的文件路径名	RFC 2132[33]
15	域名	*N*	客户端的 DNS 域名	RFC 2132[33]
16	交换（Swap）服务器	*N*	交换服务器地址	RFC 2132[33]
17	根路径	*N*	客户端的根磁盘的路径名	RFC 2132[33]
18	扩展文件	*N*	包含厂商扩展信息（可通过 TFTP 检索）的一个文件的路径名	RFC 2132[33]
19	转发开/关	1	使能/禁止 IP 报文转发	RFC 2132[33]
20	源路由开/关	1	对于指定非本地源路由的报文，使能/禁止 IP 报文转发	RFC 2132[33]
21	策略过滤器	*N*	对于指定非本地源路由的报文，为可被转发 IP 报文，指定可接受的非本地下一跳	RFC 2132[33]

（续）

代码	名称	Len（长度）	含　义	参考文献
22	最大数据报重新组装尺寸	2	客户端准备重组的数据报最大尺寸，指定为一个 16bit 的无符号整数	RFC 2132[33]
23	默认 IP TTL	1	在外发报文的 IP 首部 TTL 字段中，使用的默认 IP 存活时间数值	RFC 2132[33]
24	路径 MTU 老化超时	4	依据 RFC 1191，当执行路径最大传输单元（MTU）发现时，以 s 为单位表示的超时；MTU 发现有助于最小化路径上的报文分段	RFC 2132[33]
25	路径 MTU 平稳状态表	*N*	依据 RFC 1191，当执行路径最大传输单元（MTU）发现时，所使用 MTU 尺寸的一个表	RFC 2132[33]
26	接口 MTU	2	这个设备接口所用 MTU 的数值	RFC 2132[33]
27	所有子网都是本地的	1	指明在客户端的网络内部的所有子网是否使用本地子网（客户端连接的子网）相同的 MTU	RFC 2132[33]
28	广播地址	4	规范客户端子网所用的广播 IP 地址	RFC 2132[33]
29	掩码发现	1	规范客户端是否应该执行子网掩码发现	RFC 2132[33]
30	掩码提供者	1	规范客户端是否应该对执行掩码发现的其他客户端做出响应	RFC 2132[33]
31	路由器发现	1	规范客户端是否应该执行路由器发现	RFC 2132[33]
32	路由器请求地址	4	规范客户端应该向其直接发送路由器请求报文的 IP 地址	RFC 2132[33]
33	静态路由	*N*	规范客户端应该在其路由缓存中安装的一组静态路由；列表为“目的地网络——下一跳路由器”对（RFC 3442 定义无类静态路由选项 121 后，被废弃）	RFC 2132[33] RFC 3422[36]
34	尾部（Trailer）封装	1	规范在 ARP 消息中客户端是否应该尝试协商使用 2 层帧尾部（类似首部，但位置在帧净荷的尾端）	RFC 2132[33]
35	ARP 超时	4	ARP 缓存超时，以 s 表示	RFC 2132[33]
36	以太网封装	1	规范在一个以太网接口上客户端是应该使用以太网Ⅱ还是 IEEE 802.3	RFC 2132[33]
37	默认 TCP TTL	1	默认的 TCP 存活时间数值	RFC 2132[33]
38	TCP 保持存活（keepalive）时间	4	TCP 保持存活间隔，以 s 为单位表示	RFC 2132[33]

（续）

代码	名称	Len（长度）	含　义	参考文献
39	TCP 保持存活垃圾(garbage)	1	出于与较老实现的兼容性考虑，规范在 TCP 保持存活消息内部，客户端是否应该发送一个字节的"垃圾"	RFC 2132[33]
40	NIS 域	*N*	网络信息服务(NIS)域	RFC 2132[33]
41	NIS 服务器	*N*	*N*/4 网络信息服务服务器地址	RFC 2132[33]
42	NTP 服务器	*N*	*N*/4 网络时间服务器地址	RFC 2132[33]
43	厂商特定(信息)	*N*	厂商特定信息	RFC 2132[33]
44	NETBIOS 名字服务器	*N*	*N*/4 NETBIOS 时间服务器(即 WINS 服务器)地址	RFC 2132[33]
45	NBDD 服务器	*N*	*N*/4 NETBIOS 数据报分发(NBDD)服务器地址	RFC 2132[33]
46	NETBIOS 节点类型	1	将客户端指派为一个特定的 NETBIOS 节点类型	RFC 2132[33]
47	NETBIOS 范围	*N*	为客户端指派 NETBIOS 范围	RFC 2132[33]
48	X 窗口字体服务器	*N*	*N*/4 窗口字体服务器地址	RFC 2132[33]
49	X 窗口显示管理器	*N*	*N*/4 窗口显示管理器地址	RFC 2132[33]
50	地址请求	4	客户端请求的 IP 地址(在一条发现消息内)	RFC 2132[33]
51	地址时间	4	客户端请求的 IP 地址租赁时间(在一条发现或请求消息内)	RFC 2132[33]
52	选项过载	1	指明"sname"和/或"file"DHCP 首部字段包含其他 DHCP 选项信息，如果返回到客户端的选项超出了消息中的正常选项空间	RFC 2132[33]
53	DHCP 消息类型	1	DHCP 消息类型，见我们在本章前面讨论部分(发现，提供等)	RFC 2132[33]
54	DHCP 服务器标识符	4	(比如)为了在多项提供间做出区分，为识别服务器，在提供(Offer)(以及请求(Request)和可选的 ACK、NAK)中提供的 DHCP 服务器识别	RFC 2132[33]
55	参数列表	*N*	客户端所请求参数的 DHCP 选项代码号的列表	RFC 2132[33]
56	DHCP 错误消息文本	*N*	包含一条错误消息的文本；在一条发送到客户端的 Nak 消息中可由服务器使用，或在一条拒绝消息中由客户端使用；例如，该文本可被包含在日志细节中	RFC 2132[33]

（续）

代码	名称	Len（长度）	含 义	参考文献
57	最大 DHCP 消息尺寸	2	客户端希望接收的最大 DHCP 消息长度	RFC 2132[33]
58	刷新时间(T1)	4	从地址指派时间到客户端进入刷新状态之间的间隔	RFC 2132[33]
59	重新绑定时间(T2)	4	从地址指派时间到客户端进入重新绑定状态之间的间隔	RFC 2132[33]
60	厂商类标识符	*N*	客户端用来指派一个厂商特定的标识符	RFC 2132[33]
61	客户端 ID	*N*	客户端标识符	RFC 2132[33]
62	Netware/IP 域	*N*	Netware/IP 域名	RFC 2132[33]
63	Netware/IP 选项	*N*	Netware/IP 子选项	RFC 2132[33]
64	NIS + 域	*N*	NIS + 客户端域名	RFC 2132[33]
65	NIS + 服务器	*N*	NIS + 服务器地址	RFC 2132[33]
66	TFTP 服务器名	*N*	TFTP 服务器名；当“sname”DHCP 首部字段由其他选项过载时，可以加以使用	RFC 2132[33]
67	启动文件名	*N*	启动文件名；当“file”DHCP 首部字段由其他选项过载时，可以加以使用	RFC 2132[33]
68	家乡代理	*N*	*N*/4 移动 IP 家乡代理地址	RFC 2132[33]
69	SMTP 服务器	*N*	*N*/4 简单邮件传输协议(SMTP)服务器地址，用于外发电子邮件	RFC 2132[33]
70	POP3 服务器	*N*	*N*/4 邮箱协议(Post Office Protocol)v3(POP3)服务器地址，用于进入的电子邮件检索	RFC 2132[33]
71	NNTP 服务器	*N*	*N*/4 网络新闻传输协议(NNTP)服务器地址	RFC 2132[33]
72	WWW 服务器	*N*	*N*/4 万维网(WWW)服务器地址	RFC 2132[33]
73	IRC 服务器	*N*	*N*/4 Finger 服务器地址；finger 服务器支持基于登录名、登录时长以及其他参数来检索主机用户信息	RFC 2132[33]
74	Finger 服务器	*N*	*N*/4 因特网中继聊天(IRC)服务器地址	RFC 2132[33]
75	StreetTalk(街谈)服务器	*N*	*N*/4 街谈服务器地址；街谈是一项 Banyan Vines 用户和资源目录	RFC 2132[33]
76	STDA 服务器	*N*	*N*/4 街谈目录助理(STDA)服务器地址；街谈是一项 Banyan Vines 用户和资源目录	RFC 2132[33]

（续）

代码	名称	Len（长度）	含义	参考文献
77	用户类型	*N*	用户类型标识符	RFC 3004[38]
78	SLP目录代理	*N*+1	*N*/4服务位置协议(SLP)目录代理IP地址(可能有多个地址)	RFC 2610[39]
79	SLP服务范围	*N*	SLP代理被配置使用的SLP服务范围	RFC 2610[39]
80	快速提交	0	快速提交——针对移动性或额外负担受约束的应用而言,请求一个两报文DHCP事务,而不采用正常的四报文DORA过程	RFC 4039[40]
81	客户端FQDN	*N*	完全合格的域名(FQDN)——定义客户端的FQDN,以及客户端或DHCP服务器是否应该更新DNS	RFC 4702[41]
82	中继代理信息	*N*	中继代理信息——由中间的中继代理提供的其他客户端信息	RFC 3046[42]
83	因特网存储名服务(iSNS)	*N*	iSNS服务器地址和iSNS应用信息	RFC 4174[43]
84	未指派	—	—	RFC 3679[44]
85	NDS服务器	*N*	*N*/4要联系的Novell目录服务(NDS)服务器IP地址,目的是NDS客户端认证并访问NDS目录库	RFC 2241[45]
86	NDS树名称	*N*	客户端应该联系的NDS库的NDS树名称	RFC 2241[45]
87	NDS上下文	*N*	客户端应该使用的NDS库内部的NDS初始上下文	RFC 2241[45]
88	广播和组播服务器(BCMCS)控制器域名	*N*	BCMCS域名(FQDN)列表,用来构造接下来的SRV查询(可能是多条查询)(BCMCS被用于3G无线网络,使移动手机能够接收广播和组播服务)	RFC 4280[46]
89	BCMCS控制器IPv4地址	*N*	*N*/4 BCMCS控制器IP地址(可能是多个地址)(BCMCS被用于3G无线网络,使移动手机能够接收广播和组播服务)	RFC 4280[46]
90	认证	*N*	认证选项,依据DHCP认证协议,在客户端和服务器之间传递认证信息	RFC 3118[47]
91	客户端最近一次事务时间选项	4	在这次租赁期(依据一条DHCP租赁查询消息查询得到的数值)上与该客户端最近一次事务以来的秒数	RFC 4388[48]
92	关联的IP选项	*N*	依据一条DHCP租赁查询消息查询得到的数据,与该客户端关联的IP地址列表	RFC 4388[48]

（续）

代码	名称	Len（长度）	含　义	参考文献
93	PXE 客户端系统	*N*	PXE 客户端系统架构类型，每个类型都被编码为 16bit 代码，例如 Intel x86PC、DEC Alpha、EFI x86-64 等	RFC 4578[49]
94	PXE 客户端网络接口	3	PXE 客户端网络接口标识符，针对接口类型、接口主版本号和接口次版本号分别使用不同的字节编码	RFC 4578[49]
95	LDAP	*N*	轻量目录访问协议服务器；这个选项由 Apple 计算机使用，但没有发布指导性的 RFC	RFC 3679[44]
96	未指派	—	—	RFC 3679[44]
97	PXE 客户端机器标识符	*N*	带有编码类型和标识符数值的 PXE 客户端机器标识符	RFC 4578[49]
98	用户认证协议（UAP）	*N*	能够处理认证请求之服务的位置列表（URL），是适应开放组的 UAP 封装的	RFC 2485[50]
99	城市（civic）位置	—	服务器的位置，依据服务器提供的以国家特定城市（例如邮编）格式，最接近客户端的网元或者就是客户端自身	RFC 4776[51]
100	时区	*N*	依据 IEEE 1003.1 TZ（POSIX）编码的时区	RFC 4833[52]
101	时区数据库	*N*	参考一个本地（客户端相关的）TZ 数据库，来查询时区	RFC 4833[52]
102 ~ 111	未指派	—	—	RFC 3679[44]
112	Netinfo 地址	*N*	NetInfo 父服务器地址：虽然没有发布指导性的 RFC，但这个选项由 Apple 计算机使用；NetInfo 是用于 Apple 设备的一个分布式数据库用户和资源信息	RFC 3679[44]
113	Netinfo 标签	*N*	NetInfo 父服务器标签：虽然没有发布指导性的 RFC，但这个选项由 Apple 计算机使用；NetInfo 是用于 Apple 设备的一个分布式数据库用户和资源信息	RFC 3679[44]
114	URL	*N*	统一的资源定位器；虽然没有发布指导性的 RFC，但这个选项由 Apple 计算机使用	RFC 3679[44]
115	未指派	—	—	RFC 3679[44]
116	自动配置	1	指令客户端是否自动配置一个链路本地地址（69.254.0.0/16）。这个选项由 DHCP 服务器使用，用来通知客户端，告知 DHCP 服务器没有 IP 地址可指派，客户端可以（或不可）进行自动配置	RFC 2563[53]

（续）

代码	名称	Len（长度）	含　义	参考文献
117	名字服务搜索	*N*	以优先级顺序列出一个或多个名字服务，客户端应该用之进行名字解析：DNS、NIS、NIS + 或 WINS	RFC 2937[54]
118	子网选择	4	确定一个 IP 子网（地址），从中向这个客户端分配一个 IP 地址——覆盖 GIAddr 设置或 DHCP 服务器接口（是在该接口上接收到一条广播发现（Discover）的）	RFC 3011[55]
119	域搜索	*N*	列出一个或多个域，用于客户端解析器的配置。如果应用请求一个非 FQDN 主机名的解析，则在查询之前，这些域将顺序地添加到主机名之后。	RFC 3361[57]
120	SIP 服务器	*N*	一个或多个会话初始协议（SIP）服务器 FQDN（可能是多个）或 SIP 服务器 IP 地址（可能是多个地址）的一个列表。SIP 是多媒体呼叫或会话管理的一个控制协议	RFC 3361[57]
121	无类静态路由	*N*	规范确定客户端应该在其路由缓存中安装的一组静态路由；按照"〈CIDR 掩码长度〉.〈目的地网络〉——下一跳路由器"对的形式列出。目的地网络部分仅列出有意义的字节，丢弃本地（非子网）部分；例如 172.16.0.0/12 将被编码为 12.172.16，10.0.0.0/18 被编码为 18.10.0.0	RFC 3442[58]
122	CableLabs 客户端配置	*N*	规范确定资源（例如准备服务器、DHCP 服务器等）位置以及有线多媒体终端适配器（MTA）使用的参数，MTA 是运行在一个 DOCSIS 有线网络上的客户端设备，提供 VoIP 和有关的多媒体服务	RFC 3495[59]
123	位置配置信息（LCI）	16	为客户端提供它的 LCI，包括纬度、经度、高度以及每个坐标的分辨率	RFC 3825[60]
124	识别厂商的厂商类	*N*	使之能够规范多个厂商类，每个类都是以 IANA-指派的企业号（EN）；在支持该设备方面，这对于识别硬件厂商、软件厂商、应用厂商等方面是有用的	RFC 3925[61]
125	识别厂商的厂商特定信息	*N*	依据 IANA-指派的 EN 识别得到厂商，根据厂商对 DHCP 选项集合分组	RFC 3925[61]
126、127	未指派	—	—	RFC 3679[44]

（续）

代码	名称	Len（长度）	含 义	参考文献
128 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	Etherboot（以太网启动签名）。6 字节：E4：45：74：68：00：00			
	DOCSIS“全安全性”服务器 IP 地址			
	TFTP 服务器地址（用于 IP 电话软件负载）			
129 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	核心选项。变长字符串			
	呼叫服务器 IP 地址			
130 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	以太网接口。变长字符串			
	区分字符串（用来识别厂商）			
131 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	远端统计服务器 IP 地址			
132 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	802.1Q VLAN ID			
133 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	802.1D/p L2 优先级			
134 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	Diffserv 码点			
135 过载的（Overloaded）	PXE-未定义（厂商特定的）			RFC 4578[49]
	电话特定应用的 HTTP 代理			
136	PANA 代理	N	识别 PANA（为网络接入携带认证信息的协议）认证代理的一个或多个 IPv4 地址，针对网络接入服务，由客户端用于认证和授权	RFC 5192[62]
137	LoST 服务器	N	位置到服务转换（LoST）服务器域名；LoST 协议将服务标识符和位置信息映射到服务 URL	RFC 5223[63]
138	CAPWAP 接入控制器	N	无线接入点的控制和准备（CAPWAP）接入控制器 IP 地址（可能是多个地址），客户端可连接到这些地址	RFC 5417[64]

（续）

代码	名称	Len（长度）	含义	参考文献
139	移动性服务(MoS)IP 地址	N	提供特定类型 IEEE802.21 MoS 服务器的 IPv4 地址	RFC 5678[65]
140	MoS FQDN	N	提供特定类型 IEEE802.21 MoS 服务器的 FQDN	RFC 5678[65]
141	SIP 用户代理配置	N	会话初始协议(SIP)用户代理配置	draft-lawrence sipforum-useragent-config-03.txt[179]
142~149	未指派			RFC 3942[66]
150	TFTP 服务器地址 Etherboot(以太网启动)GRUB 配置路径名			RFC 5859[175]
151~174	未指派			RFC 3942[66]
175	以太网启动(临时指派的——2005 年 6 月 23 日)			
176	IP 电话(临时指派的——2005 年 6 月 23 日)			
177	以太网启动(临时指派的——2005 年 6 月 23 日)			
178~207	未指派的			RFC 3942[66]
208	PXE 魔数(弃用)	4	F1:00:74:7E	RFC 5071[67]
209	PXE 配置文件	N	第二阶段 PXE 启动载入的配置文件名或文件路径名	RFC 5071[67]
210	PXE 路径前缀	N	PXE 配置文件选项中指定文件名的配置文件路径前缀	RFC 5071[67]
211	PXE 重启时间	4	如果 TFTP 服务器不可达,需要等待重启的秒数	RFC 5071[67]
212	6rd 配置		6rd 顾客边缘设备配置(6rd 是一项服务提供商 IPv4-IPv6 技术——见第 15 章)	RFC 5969[176]
213	LIS 域名		这个接入网络的位置信息服务器(LIS)域名	draft-ietfgeopriv-lisdiscovery-15.txt[178]
214~219	未指派			
220	子网分配选项(临时指派的——2005 年 6 月 23 日)			
221	虚拟子网选择选项(临时指派的——2005 年 6 月 23 日)			
222~223	未指派的			RFC 3942[66]
224~254	保留的(私有用途)			
255	结尾	0	没有	RFC 2132[33]

① $N/4$ 表示法指使用“N”个字节表示一个或多个 IPv4 地址，每个地址由 4 个字节组成；因此对于长度 N，该字段将包含 $N/4$ 个完全的 IPv4 地址。当然这意味着在数据类型为 IP 地址时，N 是 4 的倍数。

② IEN－16＝因特网试验说明 16；随着 TCP/IP 在 APRANET 上投入日常使用，IEN 最终与 RFC 合并了。

4.5　动态地址指派的其他方式

虽然 DHCP 为网络管理员提供了在许多子网上预分配动态地址池的一种方式，并提供一种机制区分不同设备类型，以便执行一个 IP 地址和配置参数的区分性指派，但还存在动态地址指派的其他方法（虽然不太常用）。除了地址自动配置外，一种常见的替代方法是使用一台 Radius 服务器来指派一个 IP 地址。Radius 或后续协议 Diameter，为尝试接入一个网络的各 IP 主机提供一项认证、授权和计费（AAA）服务。当客户端尝试接入一台网络边缘设备或拨号池时，从一个客户端到一台 Radius 服务器的连接普遍通过一条点到点（PPP）或扩展的认证协议（EAP）（比如）连接实现的。Radius 服务器请（challenge）客户端输入一个用户名和口令，依据其内部或外部数据库对输入的信息进行认证，最后通过向客户端提供一个 IP 地址而提供到网络的接入。

虽然极大地简化了 Radius 协议，但这里的有关概念是某些 Radius 服务器或甚至边缘路由器设备，可被配置带有地址池，从中可向被授权的客户端实施各 IP 地址的指派。在一些情形中，Radius 服务器可被配置成实际上利用 DHCP 协议，从一台 DHCP 服务器得到一个 IP 地址。在这种情形中，Radius 服务器作为一个 DHCP 代理客户端，代表请求客户端得到一个 IP 地址，并将该地址指派给该请求客户端。我们将在第 7 章讨论一些替代的 DHCP 服务器部署策略，我们将部署在边缘设备上的 DHCP 与部署在分散 DHCP 服务器上的 DHCP 进行比较。

第5章　用于IPv6的DHCP（DHCPv6）

对于动态得到IPv6地址的那些设备而言，有两种主要策略可用于自动化这个地址指派过程：基于客户端的过程或基于网络服务器的过程。在第2章，我们以地址自动分配的形式介绍了基于客户端的地址指派概念，在该过程中，一个客户端依据路由器通告确定它的位置，并自动地计算它的接口标识符，以此推演得到一个IP地址。但是，如果在重复地址检测的前提过程中，主机确定它的自动配置的地址已被使用，那么它必须重新推演得到另一个地址或等待人工干预。

基于网络服务器的地址指派（例如DHCP）使一台主机在从IP网络中的一台服务器处请求一个IP地址（还有其他参数）过程中，宣告它的存在。用于IPv6地址的DHCP被称作DHCPv6，并在RFC 3315中定义。依据定义，DHCPv6并不与用于IPv4的DHCP集成在一起。这意味着DHCPv6仅支持IPv6地址和配置，而不附加地支持IPv4地址和参数。这留待未来的开发进行定义，如果未来需求紧迫的话。

5.1　DHCP比较：DHCPv4和DHCPv6[⊖]

与DHCPv4相比较而言，DHCPv6使用不同的消息类型和报文格式，但在许多方面是类似的。表5-1突出显示了这些相似点和差异。

表5-1　DHCPv4和DHCPv6的比较

特征	DHCPv4	DHCPv6
初始客户端消息的目的地IP地址	广播(255.255.255.255)	组播到链路范围地址:所有DHCP代理地址(FF02::1:2)
DHCP中继支持	如为"支持",则在每个中继中配置DHCP服务器地址	如为"支持",则在每个中继代理中配置DHCP服务器地址,或使用All_DHCP_Servers站点范围的组播地址(FF05::1:3)
中继代理转发	相同的消息类型码,但插入giaddr,并将之单播到DHCP服务器(可能是多台)	在发送到DHCP服务器的RELAY-FORW和来自服务器的RELAY-REPL中封装客户端消息
发送到定位服务器的消息,目的是得到IP地址和配置	DHCPDISCOVER	SOLICIT
与客户端有关的服务器消息	DHCPOFFER	ADVERTISE
接受参数的客户端消息	DHCPREQUEST	REQUEST
租赁绑定的服务器确认	DHCPACK	REPLY

⊖ 本章的开始几节依据参考文献［11］的第3章。

（续）

特征	DHCPv4	DHCPv6
为延长租赁期而发送到租赁 DHCP 服务器的客户端消息	DHCPREQUEST(单播)	RENEW(单播)
为延长租赁期而发送到任意租赁 DHCP 服务器的客户端消息	DHCPREQUEST(广播)	REBIND(组播)
放弃一个地址租赁的客户端消息	DHCPRELEASE	RELEASE
指明一个提供的 IP 地址已被使用的客户端消息	DHCPDECLINE	DECLINE
指令客户端得到一个新配置的服务器消息	DHCPFORCERENEW	RECONFIGURE
仅请求 IP 配置(不是地址)	DHCPINFORM	INFORMATION-REQUEST

5.2　DHCPv6 地址指派

当一台设备在一个 IPv6 子网上初始化时，它将倾听或请求一条路由器通告，从而确定是否有 DHCPv6 服务可用于该子网。回顾一下在第 2 章我们关于邻居发现的讨论，其中在路由器通告内部的 M 比特通知子网上的设备，告知它们，DHCPv6 服务可用于地址和参数指派；O 比特指明 DHCPv6 服务可用于参数设置，而不可用于地址指派。DHCPv6 过程是以一个客户端发出一条 SOLICIT 消息开始的，本质上从 DHCP 服务器（可能是多个服务器）处请求一个“bid”（出价，即提供一个地址），这些服务器可在该客户端所连接的特定子网上提供一个 IP 地址。在 IPv4 中客户端要广播这条起始报文，但在 IPv6 中却不是这样的，而是客户端向 All_Relay_Agents _ and _ Servers（所有中继代理和服务器）组播地址 FF02:: 1: 2 发送 SOLICIT 消息。注意在这个组播地址上的范围字段（以黑体突出显示的 **FF02**:: 1: 2）适用于链路本地范围。

在这个子网上的 DHCPv6 服务器将直接接收到 SOLICIT 报文，并以一条 ADVERTISE 报文做出响应，指明一个优先级数值。使用优先级数值的目的是使客户端选择通告最高优先级（由管理员配置）的服务器。该服务器也将指明在子网上它是否有可用的地址。如果 SOLICIT 是直接使用由 SOLICIT 报文得到的客户端源 IP 地址（极可能是客户端的链路本地地址）接收到的，那么 ADVERTISE 报文将会单播到客户端。

客户端分析接收到的各条通告，并选择一台服务器（典型情况下具有最高的优先级），请求一个 IP 地址，并向该服务器发出一条 REQUEST 消息。之后服务器将记录地址指派，并以一条 REPLY 消息对客户端做出应答，如图 5-1 所示。

在链路上被配置为中继代理的任何路由器，如果从一台 DHCPv6 客户端接收到 SOLICIT 报文，那么它将该报文中继到一个或多个 DHCPv6 服务器。IPv6 中继代理不会像在 IPv4 情形中那样要求配置 DHCP 中继代理地址，但它们可以支持这样的配置。在 IPv4 中是简单地将报文转发到一台或多台 DHCP 服务器，在 IPv6 中不是这样的，IPv6 中继代理将原始的 SOLICIT 报文封装在一条 RELAY-FORW 报文内部。之后这条报文被发送到配置好的 DHCP 服务器，或通过组播发送到范围受限的所有 DHCP 服务器组播地址（FF05:：1：3）。类似于 IPv4 DHCP GIAddr 参数，RELAY-FORW 报文的链路地址字段指明了这样的链路，客户端当前在这条链路上，且该客户端正在请求一个 IP 地址。这个过程如图 5-2 所示。这个信息由 DHCPv6 服务器使用，用于为这条链路指派一个合适的 IP 地址。DHCPv6 服务器将其 ADVERTISE 消息封装在一条 RELAY-REPL 报文中，并将之单播到相应的中继代理。

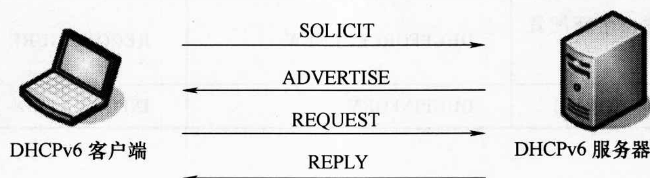


图 5-1　DHCPv6 地址指派

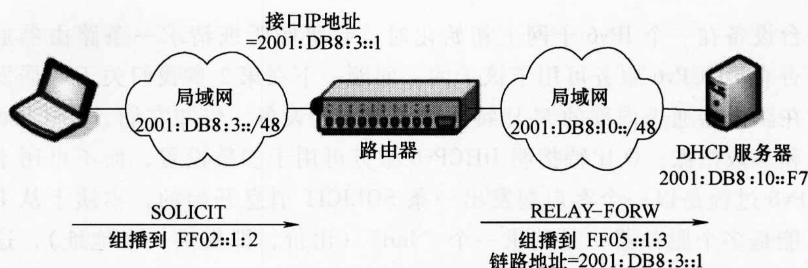


图 5-2　DHCPv6 中继[11]

当客户端接收到一条确认地址指派的应答报文时，该客户端必须执行重复地址检测，以便确保没有其他设备已在使用该 IP 地址（采用自动配置或人工配置的方法配置的）。如果另一台设备正在使用被指派的 IP 地址，那么客户端将向 DHCP 服务器发送一条拒绝（Decline）消息，指明该地址已被使用。之后该客户端重新启动 DHCP 过程，以便得到一个不同的 IP 地址。

除了上面概述的四报文交换外，DHCPv6 的特征功能还有一个快速提交选项。这使消息需求减半，使服务器能够简单地对一条 SOLICIT 报文做出 REPLY（应答）。客户端将在其 SOLICIT 消息中包括快速提交选项。可对一个地址指派做出响应的服务器（可能是多台服务器）将直接发出一条 REPLY 报文，同样包括快速提交选项。注意，做出响应的每台服务器将假定它所指派的地址将被租赁，所以快速提交应该带有短的租赁时间，或由有限数量的服务器支持，条件是正常情况下，有许多台服务器服务该子网。

和第 2 章中描述的 IPv6 自动配置一样，通过 DHCP 指派的每个非临时[⊖] IPv6 地址都有一个首选寿命和一个有效的寿命。在首选寿命超期之后，该地址被认为是有效的，但应该弃用。在弃用情况下，不应该有新的 IP 通信会话利用该地址。

5.3　DHCPv6 前缀委派

DHCPv6 不仅用于向主机指派个体（individual）IP 地址和/或关联的 IP 配置信息，而且可用于将整个网络委派给请求（地址）的路由设备。通过 DHCPv6 的这种形式委派被称作前缀委派。前缀委派的这种原始动机来自于宽带服务提供商，他们寻求以一种层次化的方式向宽带用户委派 IPv6 子网（例如/48 到/64 网络）的过程自动化。在服务提供商网络边缘的一台请求（地址）的路由器设备（面向用户，即服务用户），将通过 DHCPv6 协议向一台委派路由器发出地址空间的请求。注意该术语的含义：其意图是作为一个路由器间的协议，即使一台 DHCPv6 路由器可执行委派路由器的功能（但它不是路由器）。

前缀委派过程利用前面描述过的图 5-1 所示地址指派相同的基本 DHCPv6 消息流：索求（Solicit）、通告、请求和应答。在相应 DHCPv6 消息内部的其他信息被用于确定委派的一个合适网络。和 IP 地址一样，前缀也有首选寿命和有效寿命。发出请求的路由器可通过 DHCPv6 刷新和重绑定消息，来请求得到一次寿命延长。

5.4　DHCPv6 对地址自动配置的支持

当我们在第 2 章讨论 IPv6 自动配置时，我们定义了三种类型的自动配置：

1）无状态的。这个过程是“无状态的”，原因是它不依赖于外部指派机制（例如 DHCPv6）的状态或是否可用。

2）有状态的。有状态过程单纯依赖于外部地址指派机制，例如 DHCPv6。

3）无状态和有状态的组合。这个过程将一种形式的无状态地址自动分配与额外 IP 参数的有状态配置相结合一起使用。

自动配置的这第三种组合形式利用 DHCPv6，不是为了得到 IPv6 地址指派，而是为了得到额外参数的指派，被编码为 DHCPv6 选项。客户端可通过信息请求消息来请求配置参数，指明它正在寻求得到哪些选项参数值。能够提供所期望配置参数的一台服务器（或多台服务器）将以一条应答消息做出响应，带有相应的选项参数。

5.4.1　DHCPv6 消息类型

针对 DHCPv6 定义了如下消息类型：

1）SOLICIT（索求）——消息类型 = 1——由一个客户端发出，为的是定位 DHCPv6 服务器。

⊖　一个临时地址是一个短时（指使用时间）不可刷新的地址。

2）ADVERTISE（通告）——消息类型 =2——作为对一条索求消息的响应，由一台服务器发出，指明 DHCP 服务用服务器的存在。

3）REQUEST（请求）——消息类型 =3——由客户端发出，从一台特定的 DHCPv6 服务器请求 IP 地址和配置参数。

4）CONFIRM——消息类型 =4——由一台客户端向任何可用的服务器发出，用来验证指派给它的地址（可能是多个地址）对于它当前的子网位置仍然是合适的。

5）RENEW——消息类型 =5——由一个客户端向它所接受 IP 地址的服务器发出，用来扩展或刷新它的 IP 地址寿命，并更新其他参数。

6）REBIND——消息类型 =6——由一个客户端向所有可用的服务器发出，用来扩展它的 IP 地址寿命，并更新其他参数。是在没有接收到一条以前 RENEW 消息的应答之后，才发送这条消息的。

7）REPLY——消息类型 =7——作为对索求、请求、刷新或重新绑定消息的响应，由一台服务器发出的，用来向一个客户端提供 IP 地址和/或配置参数。服务器也向期望通过确认消息来确认其配置的客户端，发出这个消息类型，并用来确认从客户端接收到释放和拒绝消息。

8）RELEASE——消息类型 =8——由一个客户端向它接收到 IP 地址的服务器发出的，用来放弃 IP 地址。之后客户端必须释放使用该 IP 地址。

9）DECLINE——消息类型 =9——由一个客户端发出的，用来通知一台服务器，告知由该服务器指派的一个或多个地址在客户端所在的链路上已经在用。

10）RECONFIGURE——消息类型 =10——由一台服务器发出的，用来指令一个客户端重新初始化，原因是该服务器有新的或更新的配置参数可用于该客户端。之后该客户端必须按照服务器的指令，发出一条刷新或信息请求消息，来得到更新的或新的信息。

11）INFORMATION-REQUEST——消息类型 =11——由客户端发出的，用来从一台服务器得到 IP 地址之外的配置参数。

12）REPLAY-FORW——消息类型 =12——由一台中继代理直接或通过其他代理，向一台服务器或一组服务器发出的，用来封装一条客户端发起的或中继代理发起的消息。

13）RELAY-REPL——消息类型 =13——作为对 RELAY-FORW 的应答，由一台服务器向一台中继代理发出的，封装发往一个客户端的一条消息，它被编码为 RELAY-REPL 消息内部的一个选项。该中继代理必须直接或通过其他中继代理将该消息发送到该客户端。

14）LEASEQUERY——消息类型 =14——由诸如访问集中器或中继代理的一台设备发出，用于从 DHCP 服务器请求租赁绑定信息，如一个特定客户端的 IPv6 地址、DUID、中继代理、链路地址或远程标识符。IPv6 客户端 DUID 查询用于个体设备租赁查询，而其他查询类型则方便了多个客户端租赁状态的成块租赁查询。在 IETF 内部正在开发针对 IPv4 的成块租赁查询。

15）LEASEQUERY-REPLY——消息类型 = 15——作为对一条 LEASEQUERY 消息的响应，由一台服务器向查询的设备发出的，带有与查询有关的租赁绑定信息。

16）LEASEQUERY-DONE——消息类型 = 16——由一台服务器向查询的设备发出的，指明一个成块租赁查询的结束。

17）LEASEQUERY-DATA——消息类型 = 17——由一台服务器向查询的设备发出的，当一个以上客户端的数据要以这样的结果提供时，用于封装单一的 DHCPv6 客户端的租赁信息。

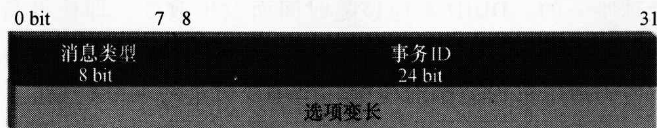


图 5-3　DHCPv6 报文格式[68]

5.4.2　DHCPv6 报文格式

DHCPv6 报文格式是非常简单的（见图 5-3）。它由一个 8bit 消息类型、24bit 事务 ID 和一个可变长度的选项字段组成。这就是报文格式，就这些内容。与客户端身份和配置有关的信息被放置在选项字段内部。

但是，当一个中继代理处于客户端和服务器之间的路径上时，该中继代理修改消息，产生用于转发和中继消息的一个通用格式，如图 5-4 所示。

1）8bit 消息类型。

2）已经中继这条消息的 8bit 跳计数或数量，沿路径由每个中继做加 1 处理。

3）128bit 链路地址——服务器使用的 IPv6 地址，用来识别客户端所处的链路（类似于 giaddr 概念）。

4）128bit 对端地址——客户端或中继代理（要被中继的消息是从该处接收到的）的 IPv6 地址。

5）可变的长度选项字段，包括中继消息选项，该选项包括要在客户端和服务器之间中继的 DHCPv6 消息

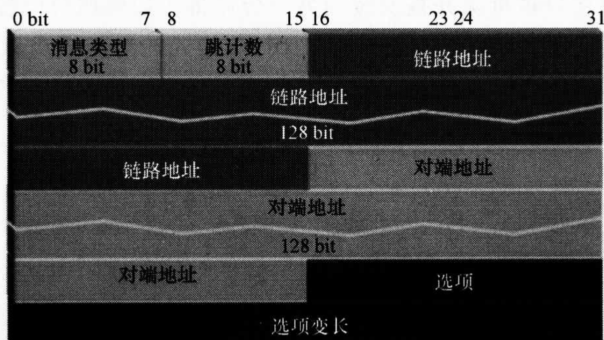


图 5-4　DHCP 中继报文格式[68]

5.5 设备唯一标识符

像 DHCPv4 一样，一台 DHCPv6 服务器必须跟踪其所配置地址池内部 IP 地址的可用性和指派情况，并识别 IP 地址的请求者和持有者。DHCPv6 利用设备唯一标识符（DUID）来识别客户端。DUID 不仅用于服务器识别客户端，而且用于客户端来识别服务器。DUID 类似于客户端-标识符概念，但 DUID 的意图为对于设备（而不是一个接口）而言是全球唯一的。DUID 不应该随时间而发生改变，即使设备的硬件发生变化时 DUID 也不应该发生变化。DUID 是以各种方式自动地由 IPv6 节点构造的。它们由一个两字节类型码后跟依据类型而变的一个可变数量的字节组成的。后跟的 DUID-类型码定义如下。

1）类型 =1——链路层地址加上时间（DUID-LLT）。

2）类型 =2——依据企业号（DUID-EN）产生的厂商指派的唯一 ID。

3）类型 =3——基于链路层的 DUID（DUID-LL）。

对于那些基于链路层地址的类型码而言，它们被用于所有的设备接口，即使由其得到链路层地址的硬件被拆除也是如此。DUID 是一个设备标识符，而不是一个接口标识符。

5.5.1 DUID-LLT

DUID-链路层地址和时间格式如图 5-5 所示。DUID 类型是“1”。硬件类型是为接口硬件类型由 IANA-指派的数值（一个完整列表参见 http：//www. iana. org/assignments/ arp-parameters）。后跟时间字段，并表示 DUID 创建的时间，以自 2000 年 1 月 1 日（UTC 时间）以来的秒数对 2^{32} 取模表示。那么所选中接口的硬件地址由链路层地址字段组成。

一台设备是如下形成这个 DUID 的：选择一个接口，使用它的链路层类型和地址。DUID 应该存储在该设备上的永久存储器之中。对于它对应的硬件类型而言，链路层地址必须是全球唯一的。之后在与 DHCP 服务器通信的过程，这同一个 DUID 与设备上的每个接口关联起来，即使 DUID 推演形成所依据的接口被拆除，也必须如此。但是，如果该接口被拆除并被安装到另一台设备，如果那台设备如此依据相同的接口地址选择形成它的 DUID 话，那么这个 DUID 格式的时间项应该使新设备使用相同接口形成不同 DUID 的概率较高。对于具有存储 DUID 的永久存储器的那些设备，建议使用 DUID-LLT 格式。

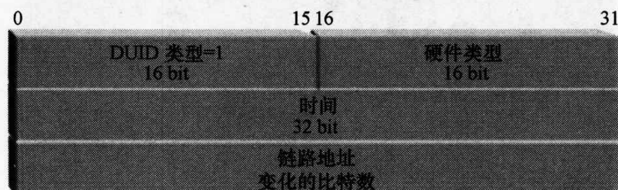


图 5-5 链路层地址加上时间格式化形成的 DUID[68]

5.5.2　DUID-EN

基于企业号码的 DUID 格式，是由厂商指派给设备的（见图 5-6）。DUID 组成包括 DUID 类型"2"、企业号码等，企业号码是由 IANA 指派给设备厂商的（参见 http：//www.iana.org/assignments/enterprise-numbers），很像由 IEEE 将以太网接口前缀指派给厂商的过程。EN 之后跟着的是由厂商指派的一个厂商唯一标识符。这个 DUID 必须被存储在设备的永久存储器之中。

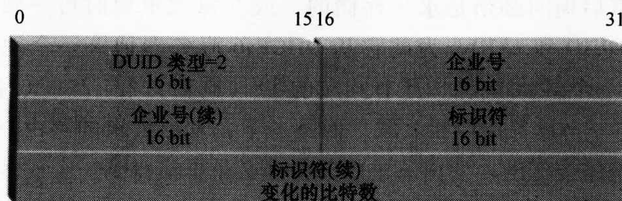


图 5-6　企业号格式化形成的 DUID[68]

5.5.3　DUID-LL

基于链路层地址的 DUID 非常类似于 DUID-LLT，但略掉了时间字段。DUID 类型是"3"（见图 5-7）。硬件类型是为接口硬件类型由 IANA-指派的值（要得到完整列表，参见 http：//www.iana.org/assignments/arp-parameters），后跟链路层地址。和其他形式的 DUID 一样，一个常见的 DUID 应该与设备上的每个接口相关联。对于没有永久存储能力来存储 DUID 值的那些设备，建议采取这种形式的 DUID。

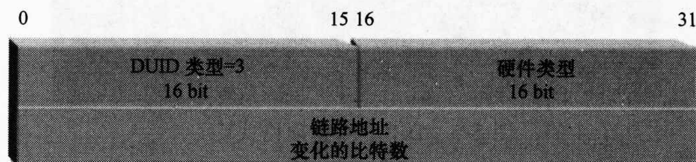


图 5-7　由链路层地址形成的 DUID[68]

5.6　身份关联

因为 DUID 与一台设备的所有接口和指派给接口的 IP 地址都关联起来了，您可能会奇怪，对于一个给定的 DUID，设备和服务器如何识别特定的接口呢。对于个体地址指派，身份关联（IA）的概念提供了一台 DHCPv6 服务器和一个客户端接口之间的这种联结关系。对于临时地址（短时租赁的、非刷新的地址）（IA_TA）、非临时地址和前缀委派（IA_PD），IA 依它们的类型区分。

临时地址指派缓解了与依据硬件地址进行自动配置地址（即修改的 EUI-64 接口 ID）（它不随时间而发生改变）相关的隐私担忧。担忧是这样的，即除非低层硬件接口发生改变，在一个 IPv6 地址内部的一个给定接口 ID 不会发生改变。因此，即使一

个设备所连接的网络天天发生变化，但接口 ID 却不变。跟踪一台设备的位置的能力，由此就会跟踪到它的用户，就变得相对容易，因此就出现了对隐私的担忧。对通过 DHCPv6 使用临时地址得到短寿命的、非刷新地址指派，是解决这种担忧的一种方法。要了解这个隐私问题的更多背景信息，请参见 RFC 3041。

对于地址指派，无论是临时的或非临时的，每个客户端接口都有一个 IA，由一个 IA 标识符加以识别（IAID）。在客户端-服务器的 DHCPv6 通信中，IAID 被表示为四个字节，并由客户端选择。在与客户端相关联的所有 IAID 中，IAID 必须是唯一的，并在客户端重启期间必须是永久存储的，或在每次重启时可一致性地推演得到。客户端指定它的 DUID 和 IAID，为此它从 DHCPv6 服务器请求一个地址。DHCPv6 服务器向 IAID 指派一个 IPv6 地址，还有相应的 T1（刷新）和 T2（重启）定时器数值。

IA_PD 不必与一个设备接口相关联。回顾一下，发出请求的路由器是使用 DHCPv6 来得到一个 IPv6 网络委派的。发出请求的路由器必须推演得到一个或多个 IA_PD，以便在 DHCPv6 内使用，IA_PD 必须在重启期间被永久存储，或可一致性地推演得到。

5.7 DHCPv6 选项

DHCPv6 选项被用于传递与所关联 DHCP 消息有关的信息，包括 DUID 和 IA。在 DHCPv6 消息内部列出各选项，并具有通用的格式，如图 5-8 所示。

在表 5-2 中给出当前定义的 DHCPv6 选项集。注意某些选项可能是内嵌的，例如与一个 IA 相关联的那些选项。



图 5-8 DHCPv6 选项格式[68]

表 5-2 DHCPv6 选项集

代码	名字	含义	参考文献
1	OPTIONS_CLIENTID	客户端标识符（客户端的 DUID）	RFC 3315[68]
2	OPTIONS_SERVERID	服务器标识符（服务器的 DUID）	RFC 3315[68]
3	OPTIONS_IA_NA	非临时地址的身份关联——包括用于非临时地址的 IAID、T1 时间、T2 时间和 IA 的其他选项	RFC 3315[68]
4	OPTIONS_IA_TA	临时地址的身份关联——包括用于临时地址的 IAID 和这个 IA 的其他选项	RFC 3315[68]
5	OPTION_IAADDR	IA 地址选项——指定 IPv6 地址和关联的首选寿命、有效寿命以及与一个 IA_NA 或 IA_TA 关联的选项。如此，这个选项仅可作为 DHCPv6 消息选项 OPTION_IA_TA 或 OPTION_IA_NA 的一个选项。	RFC 3315[68]

（续）

代码	名字	含义	参考文献
6	OPTION_ORO	选项请求的选项——由客户端使用，用于列出所请求数值的各选项代码，或由服务器用于一条重配置消息，用于指明客户端在其后续的刷新或信息请求消息中应该请求哪些选项。	RFC 3315[68]
7	OPTION_PREFERENCE	由服务器进行的优先级设置，目的是方便客户端选取 DHCP 服务器	RFC 3315[68]
8	OPTION_ELAPSED_TIME	自客户端开始当前 DHCP 事务以来的时间量，以百分之一秒为单位表示。要求客户端使用这个选项	RFC 3315[68]
9	OPTION_RELAY_MSG	由一台中继代理中继的 DHCP 消息	RFC 3315[68]
10	未指派	—	—
11	OPTION_AUTH	认证信息，用于可靠地识别一条 DHCP 消息的源，并验证消息完整性	RFC 3315[68]
12	OPTION_UNICAST	服务器单播选项，指明客户端可使用该 IP 地址向这台服务器单播消息	RFC 3315[68]
13	OPTION_STATUS_CODE	状态代码选项，指明一个 2 字节的状态码和可变长度的消息。这个选项可用作一个 DHCP 消息选项，或作为另一个 DHCP 消息选项内部的一个选项	RFC 3315[68]
14	OPTION_RAPID_COMMIT	快速提交选项，使一台客户端请求带有一个 IP 地址和参数的一条直接应答，旁路了通告和请求消息	RFC 3315[68]
15	OPTION_USER_CLASS	用户类选项——类似于 DHCPv4 中的用户类，用于协助服务器做出地址指派决策	RFC 3315[68]
16	OPTION_VENDOR_CLASS	厂商类选项——类似于 DHCPv4 中的厂商类，用于传递设备或接口的厂商或制造商信息，协助服务器做出地址指派决策。厂商类选项包括厂商的 IANA 指派的企业号码	RFC 3315[68]
17	OPTION_VENDOR_OPTS	厂商特定的信息——这个选项包括 IANA 指派的企业号码以及一个或多个选项，每个选项都以选项码、长度和值定义	RFC 3315[68]
18	OPTION_INTERFACE_ID	接口 ID 选项——由中继代理使用，用于传递在其上接收到客户端消息的代理的接口 ID。这个选项仅出现在 RELAY-FORW 消息中，且当确实出现时，服务器将其复制到 RELAY-REPL 消息上	RFC 3315[68]

（续）

代码	名字	含义	参考文献
19	OPTION_RECONF_MSG	重新配置消息选项，用于重新配置消息中，用来通知客户端重新配置要使用的消息类型；消息类型是刷新或信息-请求	RFC 3315[68]
20	OPTION_RECONF_ACCEPT	重新配置接受选项——如果客户端乐意接受来自服务器的重新配置消息，则客户端使用这个选项	RFC 3315[68]
21	OPTION_SIP_SERVER_D	SIP 服务器域名选项，列出客户端可以使用的 SIP 外发代理服务器的域名	RFC 3319[69]
22	OPTION_SIP_SERVER_A	SIP 服务器 IPv6 地址选项，列出客户端可以使用的 SIP 外发代理服务器的 IPv6 地址	RFC 3319[69]
23	OPTION_DNS_SERVERS	DNS 递归名字服务器选项——以优先级顺序列出 DNS 递归名字服务器的 IPv6 地址（可能有多个地址），客户端解析器可向其发送 DNS 查询	RFC 3646[70]
24	OPTION_SIP_LIST	域搜索列表选项——当通过 DNS 解析主机名时，为客户端用途提供一个域搜索列表	RFC 3646[70]
25	OPTION_IA_PD	前缀委派的身份关联——包括 IAID、T1 时间、T2 时间以及 IA_PD 的其他选项（包括像选项代码 26 所定义的关联前缀（可能有多个前缀））	RFC 3633[71]
26	OPTION_IAPREFIX	IA_PD 前缀选项——指定与 IA_PD 关联的 IPv6 前缀，还有关联的选项以及首选寿命和有效寿命。这个选项仅可作为 DHCPv6 消息选项 OPTION_IA_PD 的一个选项出现。这个选项被指定带有一个 8bit 前缀长度和一个 128bitIPv6 前缀	RFC 3633[71]
27	OPTION_NIS_SERVERS	网络信息服务（NIS）服务器——依据可用于 IPv6 地址而排序的 NIS 服务器列表	RFC 3898[72]
28	OPTION_NISP_SERVERS	网络信息服务 v2（NIS +）服务器——依据可用于 IPv6 地址而排序的 NIS + 服务器列表	RFC 3898[72]
29	OPTION_NIS_DOMAIN_NAME	网络信息服务域名——可被客户端使用的 NIS 域名	RFC 3898[72]
30	OPTION_NISP_DOMAIN_NAME	网络信息服务 v2（NIS +）域名——可被客户端使用的 NIS + 域名	RFC 3898[72]

（续）

代码	名字	含义	参考文献
31	OPTION_SNTP_SERVERS	简单网络时间协议（SNTP）服务器——依据可用于 IPv6 地址而排序的 SNTP 服务器列表	RFC 4075[73]
32	OPTION _ INFORMATION _ REFRESH_TIME	信息刷新选项——指定从当前时间开始秒数或数值上界，指一个客户端在从 DHCPv6 服务器接收到刷新信息之前必须等待的时间，特别对于无状态 DHCPv6 场景尤其要遵守这个时间	RFC 4242[74]
33	OPTION_BCMCS_SERVERS_D	广播和组播服务（BCMCS）域名列表——对应于 BCMCS 服务器（可能有多台）的一个或多个 FQDN 列表（BC-MCS 用于 3G 无线网络中，使移动终端可接收广播和组播服务）	RFC 4280[46]
34	OPTION_BCMCS_SERVERS_A	广播和组播服务 IPv6 地址列表——对应于 BCMCS 服务器（可能有多台）的一个或多个 IPv6 地址列表（BCMCS 用于 3G 无线网络中，使移动终端可接收广播和组播服务）	RFC 4280[46]
35	未指派	—	—
36	OPTION_GEOCONF_CIVIC	地理位置，以市政（例如邮政）格式表示。这个选项可由服务器提供，将服务器的位置、最近的网元（例如路由器）与客户端或客户端自身相关联。位置信息包括一个 ISO 3166 国家代码（US、DE、JP 等）和国家特定的位置信息，例如州、省、乡、市、街区、街组（group of streets）等	RFC 4776[51]
37	OPTION_REMOTE_ID	中继代理远端 ID 选项——中继代理在发往 DHCPv6 服务器的 RELAY-FORW 消息中插入的远端身份。在服务提供商环境中这是有用的，其中在将消息中继到 DHCPv6 服务器之前，面向订户设备的“边缘”设备为订户连接插入一个标识符	RFC 4679[75]
38	OPTION_SUBSCRIBER_ID	中继代理订户 ID 选项——中继代理在发往 DHCPv6 服务器的 RELAY-FORW 消息中插入的订户身份。在服务提供商环境中这是有用的，其中在将订户所发的消息中继到 DHCPv6 服务器之前，面向订户设备的“边缘”设备为订户插入一个标识符	RFC 4580[76]

（续）

代码	名字	含义	参考文献
39	OPTION_CLIENT_FQDN	FQDN 选项——指明客户端或 DHCP 服务器是否应该以对应于所指派 IPv6 地址的 AAA 记录和本选项中提供的 FQDN 来更新 DNS。DHCP 服务器总是更新 PTR 选项	RFC 4704[77]
40	OPTION_PANA_AGENT	这个选项提供了与 PANA（用于携带网络接入认证信息的协议）认证代理（一个客户端可以使用的）关联的一个或多个 IPv6 地址	RFC 5192[62]
41	OPTION_NEW_POSIX_TIMEZONE	由客户端使用的时区（TZ），以 IEEE 1003.1 格式表示（POSIX——便携的操作系统接口）。这种格式支持时区和夏令时间信息的文本表示	RFC 4833[52]
42	OPTION_NEW_TZDB_TIMEZONE	由表项名索引的时区数据库表项。客户端必须有 TZ 数据库的一个拷贝，它查询对应的表项，来确定它的时区	RFC 4833[52]
43	OPTION_ERO	中继代理应答（echo）请求选项——在 RELAY_FORW 消息中由中继代理用来请求 DHCPv6 服务器回应某些被请求的中继代理选项，即使服务器上不支持该选项也要回应（DHCPv4 服务器总是回应中继代理选项（82）选项，但这点在 DHCPv6 中不作要求，因此中继代理的这个选项要求这种回应）	RFC 4994[78]
44	OPTION_LQ_QUERY	查询选项用于 LEASEQUERY 消息，用来识别正被请求的查询信息。这个选项包括查询类型（由 IA 地址或客户端 ID 选项指明）、查询所施用的链路地址和查询选项	RFC 5007[79]
45	OPTION_CLIENT_DATA	客户端数据——这个选项包含针对一条 LEASEQUERY_REPLY 消息内被请求的客户端数据的查询响应信息。在最低限度情况下，这个选项包括客户端标识符（OPTION_CLIENTID）、IA 地址或前缀（OPTION_IAADDR 和/或 OPTION_IAPREFIX）和客户端发生最近一次事务的时间（OPTION_CLT_TIME）	RFC 5007[79]
46	OPTION_CLIENT_TIME	客户端最近一次事务的时间——指明自服务器最近一次与客户端（由租期查询索引标明）通信以来的秒数。这个选项被封装在一条 LEASEQUERY-REPLY 消息内部的 OPTION_CLIENT_DATA 选项内。	RFC 5007[79]

（续）

代码	名字	含义	参考文献
47	OPTION_LQ_REPLY_DATA	中继数据——用于一条 LEASE-QUERY-REPLY 消息，提供与所请求的客户端信息关联的中继代理信息。这个选项包括所接收客户端之中继信息的中继代理地址以及完整的被中继消息	RFC 5007[79]
48	OPTION_LQ_CLIENT_LINK	客户端链路——识别一条或多条链路，被查询的客户端在这些链路上具有 DHCPv6 绑定。可以地址或客户端 ID 识别被查询的客户端	RFC 5007[79]
49	OPTION_MIP6_HNINF	移动 IPv6 归属网络信息——客户端用之向服务器标识其目标归属网络（在一条信息请求消息之中）	draft-ietf-mip6-hiopt-17.txt[80]
50	OPTION_MIP6_RELAY	移动 IPv6 中继代理——由一台中继代理使用，通过一条 RELAY-FORW 消息识别归属网络信息	draft-ietf-mip6-hiopt-17.txt[80]
51	OPTION_V6_LOST	服务转换定位（LoST）服务器域名；LoST 协议将服务标识符和位置信息映射到服务 URL	RFC 5223[63]
52	OPTION_CAPWAP_AC_V6	无线接入点控制和准备（CAPWAP）接入控制器 IPv6 地址（可能有多个地址），客户端可连接这些地址	RFC 5417[64]
53	OPTION_REPLAT_ID	DHCPv6 成批租赁查询——为一个指定的中继代理（在这个选项中由其 DUID 识别）请求租赁和前缀委派绑定	RFC 5460[81]
54	OPTION_IPv6_Address-MoS	提供特定类型 IEEE 802.21 移动性服务（MoS）的服务器的 IPv6 地址（可能有多个）列表	RFC 5678[65]
55	OPTION_IPv6_FQDN-MoS	提供特定类型 IEEE 802.21 移动性服务（MoS）之服务器的 FQDN（可能有多个）列表	RFC 5678[65]
56	OPTION_NTP_SERVER	网络时间协议（NTP）和简单 NTP（SNTP）服务器地址（可能有多个）和/或域名	RFC 5908[180]
57	OPTION_F6_ACCESS_DOMAIN	在这个接入网络上位置信息服务器（LIS）的域名	draft-ietfgeopriv-lisdiscovery-15[178]
58	OPTION_SIP_UA_CS_LIST	会话初始协议（SIP）用户代理配置	draft-lawrencesipforum-useragent-config-03[179]
59	OPT_BOOTFILE_URL	客户端启动文件的 URL	draft-dhcdhcpv6-optnetboot-10[181]

（续）

代码	名字	含义	参考文献
60	OPT_BOOTFILE_PARAM	客户端启动文件参数	draft-dhcdhcpv6-optnet-boot-10[181]
61	OPTION_CLIENT_ARCH_TYPE	客户端系统架构	draft-dhcdhcpv6-optnet-boot-10[181]
62	OPTION_NII	统一网络设备接口（UNDI）支持的客户端网络接口	draft-dhcdhcpv6-optnet-boot-10[181]
63 ~ 255	未指派	—	—

第6章　DHCPv6的各项应用

DHCP的最基本应用是地址指派的自动化。当我们连接到一个IP网络时，我们想当然地使用DHCP。通过自动地进行IP层的初始化，这项基本功能使各项IP应用比较容易使用。端用户不需要呼叫（为计算机用户提供的）网络支持服务来得到IP地址，并将IP地址输入到他们的设备之中。DHCP不仅使IP地址指派自动化，而且使网络管理员保留了如下控制能力，即控制哪些IP地址可指派到某些客户端，甚至像我们将在第8章中描述的那样拒绝访问。除了基本的地址指派服务外，在本章我们将讨论依赖于DHCP的各项技术应用。当然，依赖于DHCP的这些应用因此也要依赖于与IP地址规划一致的DHCP配置。

本章突出要求特定用途DHCP配置的那些应用，这些配置包括设备特定的配置和宽带信息准备提供（服务）。基于DHCP的访问控制也可归组在这个话题之下，但我们将在第8章安全上下文下讲解那项内容。

支持采用DHCP的各项应用的基石，是DHCP服务器对请求一个地址的设备进行分类，并提供一个合适的IP地址和其他配置信息的能力。这种将客户端分类为客户端类（client class）的做法使DHCP管理员能够识别在一个特定DHCP报文字段或选项内部的一个参数值，以便在依据DHCP事务的基础上进行匹配。当一个客户端被分类时，那么DHCP服务器可确定如下内容。

1）从哪个IP地址池中向客户端指派一个地址（如果还有可用地址的话）。

2）向客户端提供哪些其他的或替代的选项参数值。

来自因特网系统联盟（ISC，Internet Systems Consortium）和微软的领先DHCP参考实现，都支持厂商类标识符（对于IPv4是选项60，对于IPv6是选项16）和用户类标识符（对于IPv4是选项77，对于IPv6是选项15）选项作为类参数。当这些选项被包括在发现（Discover）或索求（Solicit）报文中时，服务器可使用这个信息来识别请求其配置的设备的类型。

6.1　多媒体设备类型特定配置

迄今为止我们使用的最常见范例应用是多媒体设备初始化应用，例如IP上的语音（VoIP）设备。在许多情形中，多媒体厂商制造商对一个给定厂商类标识符选项值进行编码。多数厂商在厂商类标识符选项字段内部，提供一个模型号和/或制造商名。将DHCP服务器配置可识别这个特定值，就使服务器能够提供客户端要求的某些DHCP选项，并从一个特定地址池中指派一个IP地址。要求使用特定配置参数的其他针对应用的DHCP客户端，可类似地加以识别并在对应厂商类选项的值基础上进行配置。

用户类标识符选项是用来确定客户端配置的另一个候选方法。但是，典型情况下，因为用户类标识符是端用户可设置的，所以人们认为它是不太可靠的。如果用户类组之外的一名用户发现了对应于用户类组的值或设置，则他或她可相应地对他或她的设备进行编程。例如，使用微软的带有/setclassid 参数的 ipconfig 工具，要设置用户类标识符选项的值是非常容易的。

在第 4 章，当讨论 IPAM 全球公司的旧金山办事处的客户端类设置时，为了依据厂商类来区分 VoIP 设备，我们介绍了一个范例 VoIP 应用配置。图 4-5b 在这里重画为图 6-1，形象地展示了配置一台 ISC DHCP 服务器的一个简单范例，该例中说明如果一条 DHCP 报文包含了值为“vendorY”的一个厂商类标识符选项，而服务器识别类“vendor-y”的各客户端。一旦被分类为一台 vendor-y 设备，则将从带有相应路由器和 DNS 服务器选项的 10.16.129.20～10.16.129.250 池中，向客户端发行指派一个地址。指定这些选项值，在这个地址池声明内部带有“vendor-y”语句的被允许成员。

类似地，类“vendor-x”的设备将被提供值为“vendorX”的一个厂商类标识符选项的客户端加以辨认。将从 10.16.128/23 子网上带有路由器（和 tftp-server-name）选项值的 10.16.128.20～10.16.128.250 池中向这些设备指派地址。

ISC DHCP 服务器支持在其他类参数上的过滤操作，事实上，从 MAC 地址、MAC 地址的一个子网或任何选项值的任何一个报文参数均可。如果需要过滤一个给定的 MAC 地址（接口卡）或 MAC 前缀（制造商），并指派某些参数的话，则这样做是非常方便的。

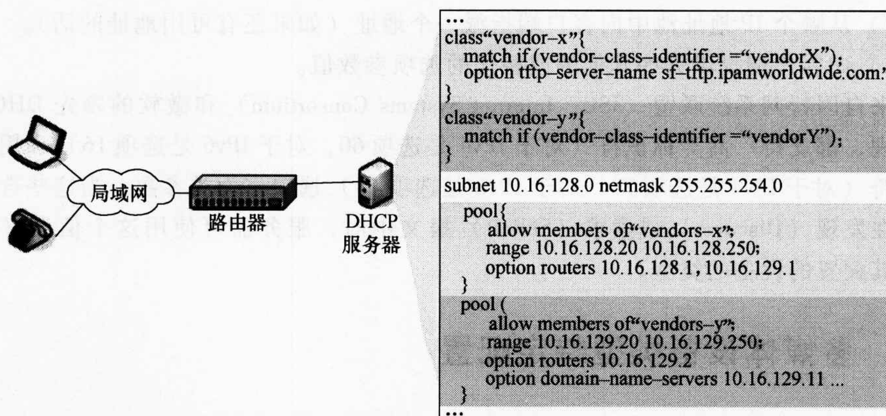


图 6-1　依据类，为 DHCP 客户端指定配置信息（依据参考文献［35］的语法）

6.2　宽带订户配置信息准备

有线电缆产业为有线电缆上的数据传输定义了一个标准，被称作有线电缆上数据服务接口规范（DOCSIS®）。DOCSIS 规范，是由 Cablelabs 撰写的，要求使

用DHCP为顾客端设备（CPE）（例如线缆调制解调器和电话设备）提供配置信息。提供有线电缆数据或宽带因特网服务的一个有线电缆运营商，必须部署DH-CP服务器来支持CPE配置信息准备过程。其他宽带技术（如数字用户线（DSL）和光纤）也可使用DHCP或Bootp，虽然诸如PPP（点到点协议）也可由这些宽带技术所用。

将DHCP集成到配置信息提供过程，这使在IP地址指派和容量以及CPE用于初始化的其他配置参数上的宽带运营商控制，是经济上负担得起的。DHCP也可被用来从对应于各种服务等级（依据客户的订购信息）的地址池中指派IP地址。从一个给定地址池中指派一个地址的做法，要求网络路由基础设施被配置成：将带有这种地址的IP报文仅路由到某些网络，允许对某些目的地的访问，并以相应的优先级和排队来处理报文。

让我们考虑一个范例来说明这些概念。在图6-2中，三个订户通过宽带接入网络被连接到同一个宽带网关上。该图将每个订户图示为带有各种服务等级，由不同的阴影表示，这些订户被连接到宽带网关的各不同端口上。取决于宽带接入技术，这些端口可能是物理端口也可能是共享网络接入的逻辑端口。

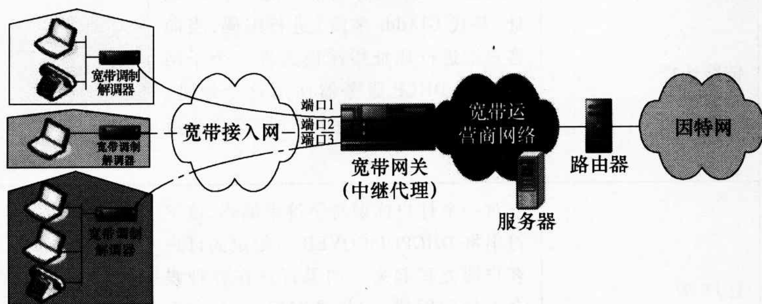


图6-2 宽带接入场景[11]

不管宽带接入技术为何种技术，使用DHCP的各服务提供商都需要依据已知的或可信的信息，进行地址和参数指派。服务提供商并不依赖于DHCP报文的客户端硬件地址字段（它可能被伪造），而是依赖于来自宽带网关的信息，该网关位于服务提供商的网络内，并被认为是值得信任的。

作为一个DHCP中继代理的宽带网关，将DHCP报文单播到合适的DHCP服务器（可能是多台），在DHCP报文首部内部插入GIAddr字段。网关在空选项终结符之前，插入中继代理信息选项参数作为最后一个选项。中继代理信息选项提供诸如订户设备硬件地址或订户虚电路标识符等信息，帮助DHCP服务器识别发出DHCPDISCOVER报文的订户客户端。

这使DHCP服务器在其配置的基础上，向一个给定的订户提供合适数量的IP地址和/或选项参数。在IPv4中的中继代理信息选项（选项82）是由一个或多个子选项组成的，如下定义这些子选项。

子选项代码	名称	描　述	RFC 索引
1	电路 ID	对有关到订户的连接信息进行编码。这由对应于订户的一个虚电路标识符(典型地对应于一个层 2 标识符,如一个 ATM 虚电路 ID、帧中继数据链路连接标识符(DLCI)),或远端接入服务器或交换机端口号组成	3046[42]
2	远端 ID	就远端客户端设备的信息进行编码,例如其以太网地址、调制解调器标识符或一条拨号连接的呼叫者 ID	3046[42]
3	保留	未使用	—
4	DOCSIS 设备类	就有线电缆 CPE 的 DOCSIS 设备类进行编码。这个选项适用于 DOCSIS 有线电缆接入网络,CMTS(有线电缆边缘设备)可在这个信息(在 DOCSIS 注册过程中采集到的)的基础上包括这个子选项	3256[82]
5	链路选择	对由 DHCP 服务器使用的一个 IP 地址(替代 GIAddr 字段)进行编码,当向客户端进行地址指派而选择一个子网地址时,DHCP 服务器使用这个地址。当正在使用共享的子网㊀时,这将是适用的。	3527[83]
6	订户 ID	对一个订户标识符字符串编码,该字符串将 DHCPDISCOVER 与给定的订户客户端关联起来。如果订户在各种媒介上访问网络,这将是有用的,其中电路标识符或远端标识符的用途将仅指明低层的接入机制,而不指明订户关联	3993[84]
7	RADIUS 属性	依据 RADIUS 协议(RFC 2865)对 RADIUS 属性编码,在进行参数指派时,DHCP 服务器将用这些属性。这些属性可被编码为一个类型长度值的字节流,并可包括用户名、口令、接入服务器 IP/端口以及其他属性	4014[85]
8	认证	对认证信息进行编码,作为在中继代理信息上提供消息完整性的一种方法。这种编码类似于第 8 章讨论的 DHCP 认证所用的编码方法	4030[86]

㊀ 共享的子网指在单一物理子网(路由器接口)上提供多个逻辑子网。

（续）

子选项代码	名称	描述	RFC 索引
9	厂商特定信息	编码为一个或多个厂商特定的信息集合，每个集合由一个三元组组成：IANA 注册的企业号、长度和数据	4243[87]
10	中继代理标志	标志条件的可扩展子选项；定义了一个标志，指明中继代理是通过单播（1）还是广播（0）接收到 DHCP 报文的	5010[88]
11	服务器标识符重写	指令 DHCP 服务器在其响应客户端时，要在服务器标识符字段中使用这个指定的值；这使中继代理能够接收 DHCPRENEW 报文（以其他方式中继代理是看不到这种报文的），当向服务器转发 DHCPRENEW 报文时，使中继代理插入与客户端关联的其他中继代理子选项值	5107[89]

在 DHCPv6 内部，定义了两个类似的选项。

1）代码 37 = Option_remote_id（选项-远端 ID）。

2）代码 38 = Option_subscriber_id（选项-订户 ID）。

让我们考虑使用 ISC DHCP 语法（35）的一个范例 DHCP 服务器配置，来形象地说明中继代理处理。这个陈述声明了类“broadband”（宽带），它依据的是中继代理识别选项的电路 ID 子选项。这里，我们定义单一客户端类，但提供子类来识别宽带类的特定实例。在这个情形中，我们为电路 ID 子选项的两个对应值简单地定义两个子类。

```
class “broadband” {
        match option agent. circuit-id;
}
subclass “modem” “45023” {
        [ declarations and parameters for modem devices ] / * 调制解调器设备的声明和
                                                          参数 * /
}
subclass “phone” “67032” {
        [ declarations and parameters for phone devices ] / *  电话设备的声明和参数 * /
}
```

一种比较具有扩展能力的方法将是利用 ISC DHCP 实现的类衍生特征。和限制租赁或可指派给一名订户的 IP 地址数量的能力一起，我们来形象地说明这点。一个基本层次的服务可承诺单一 IP 地址，而一个较高层次的服务（以及也许还有价格）可包括两个或更多个 IP 地址。lease limit（租期限制）语句支持在 ISC DHCP 配置文件内部的这项特征控制。这个语句可与一个客户类定义关联，来指定最大租期数，可用

于提供给匹配这个类的各客户端。

类衍生（spawning）功能可依据 DHCP 报文中的信息，使客户端子类的在线动态生成或产生成为可能。spawn with（以…产生）声明以产生所依据的参数定义了一个产生类。例如，DHCP 服务器可被配置为：依据每个唯一的电路 ID 中继代理子选项值来产生客户端类。因此，当 DHCP 服务器接收到一条 DHCPDISCOVER 报文时，它就分析电路 ID 子选项。如果对于给定值，存在一个类（以前产生的），则为进行处理而分析相应的参数和声明；如果不存在带有那个电路 ID 的一个类，则 DHCP 服务器为给定值产生一个新的子类。如下范例形象地说明了带有一个衍生子类的一个宽带客户端类的定义，它依据的是电路 ID，该定义使用 ISC DHCP 语法（35），将待定的订户租期限制为最大为 6。

```
class “broadband” {
    spawn with option agent. circuit-id;
    lease limit 6;
}
```

6.3 有关租期指派或限制的各项应用

依据中继代理信息，使用租期限制和参数设置的方法，并不仅适用于宽带环境。其他应用也可使用相同的技术，前提是中继代理支持中继代理信息选项的全体。在这种情形中，使用 ISC DHCP 服务器的方法，支持依据所定义类和中继代理信息参数，进行地址和参数指派以及租期限制。这项技术可用来限制某些子网上的地址指派速度，或在工厂或类似应用中向设备提供配置参数。

6.4 预启动执行环境客户端

预启动执行环境（PXE 或“Pixie”）客户端是这样的设备，它依赖于网络服务器而不是一块共存的硬盘来启动。这种无盘服务器和其他这种设备典型地使用 DHCP 来得到一个 IP 地址和启动参数（包括启动服务器地址和启动文件名）。DHCP 提供了一种简便的机制，在没有人工介入的情况下，初始化这些设备。从历史角度而言，DHCP 服务器必须配置每个 PXE 客户端的 MAC 地址，以便提供特定于该设备的配置信息，即使相同“type”（类型）的多个 PXE 客户端可准确地利用相同启动信息时也是如此。

RFC 4578[49]是一个信息型的 RFC，它定义了这样一种方式，其中一个 PXE 客户端可向服务器标识它的类型或架构。这个信息可被 DHCP 服务器用来识别并提供合适的设备初始化参数。DHCP 服务器将需要配置成匹配特定的客户端提供的 PXE 选项值，之后将这些值映射到配置参数或选项的一个对应集合，并将之返回给客户端。很自然地，这由使用客户端类处理来完成的。

可包括在 PXE 客户端和 DHCP 服务器之间的选项如下。

（1）选项 93——客户端系统架构类型——指定 PXE 设备的架构类型，并必须将其包括在事务过程中的所有 DHCP 报文之中。

1）Intel x86PC。

2）NEC/PC98。

3）EFI Itanium。

4）DEC Alpha。

5）Arc x86。

6）Intel Lean Client（瘦客户端）。

7）EFI IA32。

8）EFI BC。

9）EFX Xscale。

10）EFI x86-64。

（2）选项 94——客户端网络接口标识符——识别网络接口类型和版本，并必须被包括在事务的所有 DHCP 报文中。唯一定义的接口类型是用于统一网络设备接口（UNDI）的类型。

（3）选项 97——客户端机器标识符——识别机器启动的类型。这个选项采用一个类型和标识符进行编码。唯一定义的类型 0，指明该标识符被编码为一个 16 字节的全局唯一标识符（GUID）。

（4）选项 128～135——PXE 客户端请求这些选项，如果需要的话，其意图是用于下载的启动程序，虽然后来并没有被广泛指派为 PXE 用途。

要小心注意的是，使用选项 128～135 的 PXE 客户端可能与汇总于第 4 章中这些选项的其他被指派含义相冲突。

6.4.1 PPP/RADIUS 环境

RADIUS（远程接入拨入用户服务）协议提供了认证尝试连接到一个网络的端用户的一种方法。RADIUS 是 802.1X 的一个重要组成，802.1X 是在主要的网络接纳控制（NAC）文献内提出的一个流行的层 2 媒介接入控制协议。RADIUS 在层 3 也起了一定作用，特别当与 PPP 连接一起使用时更是如此，普遍情况下与拨号或 DSL 连接一起使用。

当在层 3 工作时，一些 RADIUS 服务器可被配置成向 PPP 连接另一端的每个客户端指派 IP 地址。这个地址指派过程可由服务器上直接配置的一个地址池完成，或配置 RADIUS 服务器通过一台 DHCP 服务器得到一个地址来完成。在后一种场景中，RADIUS 服务器的功能是代表客户端的一个 DHCP 代理。RADIUS 服务器发起 DHCP D-O-R-A 过程，发出一条 DHCPDISCOVER 报文。采用这种方法的一个说明是，RADIUS 服务器必须代表每个客户端（目的是唯一地识别这些客户端），产生一个硬件地址或客户端标识符。否则，将会使用 RADIUS 服务器的硬件地址，这时 DHCP 服务器将假定同一客户端会不断地重启，并在所有请求上指派同一 IP 地址。RADIUS 服务器可使用一种内部机制来伪造客户端的硬件地址，但需要将推演得到的地址映射到端

客户端，以便处理如刷新和释放等后续租赁事务。另一种方法是利用前面描述的中继代理信息选项的 RADIUS 属性子选项，以便唯一地识别每个客户端。

6.4.2 移动 IP

移动 IP 为一台 IP 设备保留网络连通性，同时在一个本地或远端 IP 网络周围到处移动，提供了一种机制。这种移动可能发生在从一个总部会议到在一个分支办事处打开一个新会话的一个通信会话过程中，即不仅当发起实施一个会话及之后终止该会话（比如）时会发生移动。移动设备有一个家乡地址（对应于其家乡网络），还有一个转交地址，这是在服务网络上得到的（取决于移动设备当前连接在哪儿）。例如，如果当到城镇外时，我打开我的个人数字助理（PDA）设备，我会从一个不同于我正常在家时所用的一个服务提供商处得到无线服务。只要我的家乡提供商与我正在访问的提供商有服务协议（agreement），我就应该能够手工配置、通过 DHCP 或通过自动配置得到一个地址。

IP 移动性在 IPv4 和 IPv6 之间多少存在不同，但这两个协议都利用了如下概念，即一个移动节点处理一个家乡地址（这是在家乡网络上的节点的地址）和一个转交地址（它在拜访网络上的地址）。虽然严格意义上说它不是一项 DHCP “应用”，但我们在这里提到它，是就地址分配和指派策略而言，应该是需要考虑的一个领域，并不涉及在您所在网络上拜访节点的访问安全性问题。

第7章　DHCP服务器部署策略

本章详细研究DHCP的部署策略和折中考虑。多数折中考虑会遇到预算资金和服务器数量的陷阱，所以最普遍的目标是将DHCP服务器部署到端用户将总能以及时的方式得到这些服务的位置，同时使花在部署服务器和相关联服务器生命周期间成本的总资金最小化。这个简单陈述的目标意味着对高可用和合理性能服务的需求，这些都要在预算约束之内提供。预算资金必须不仅要计算服务器购买费用，而且要计算将来的支持和维护费用，这包括服务器硬件升级、操作系统（OS）补丁和升级以及新功能、缺陷修正或安排措施的DHCP升级。

7.1　DHCP服务器平台

DHCP服务器可部署在物理硬件服务器或仪器的各种平台上或部署为一个虚拟机（VM）平台上的虚拟服务器。当我们讨论部署可能选择项时，我们比较一般化地使用“平台”这个术语，在每种情形中它通常被解释为这些选项之一。

7.1.1　DHCP软件

部署DHCP服务器的传统模型需要部署一台物理服务器（支持建议采用的处理组件）和操作系统（由相应的DHCP软件厂商支持）。为了最大化硬件利用率，在这样的服务器上也可安装其他应用。

7.1.2　虚拟机DHCP部署

存在可用于主要Windows和Linux操作系统（OS）各版本的虚拟机（VM），这使在微软VM上部署微软DHCP以及在Linux VM上部署ISC成为可能。在VM上部署DHCP的做法，节省了硬件成本、机架空间和电源走线（draw），同时相比于在一台通用硬件服务器上安装一个DHCP守护进程的做法，具有更好的隔离能力。主要的仪器设备厂商也将他们的仪器设备产品以虚拟机方式提供，这将VM的优势与仪器设备的优势结合起来，下面会讨论这点。

7.1.3　DHCP仪器设备

DHCP仪器设备是预装DHCP服务于安全的硬件平台上，典型见到的是基于Intel的带有硬化（hardened）Linux操作系统的平台。就像路由器一样，它们初始情况下是部署为运行于通用硬件上的软件，之后演化为特殊用途的硬件平台，DHCP仪器设备提供了DHCP服务自包含硬件平台的一条演进路径。仪器设备被“硬化”是指，安装在平台上的基本Linux内核被剥除了任何不必要的服务。这样得到的是一个定制

化的内核和 OS，它仅支持 DHCP 服务（以及由厂商支持的其他服务，例如 DNS）。仪器设备厂商也应该相应地也削减掉了底层文件系统、用户、权限和网络端口。

仪器设备提供一站式购买的简化部署方法，而不是不得不协调并采购服务器硬件、安装合适的 OS 版本和对应的补丁（patch level），之后安装 DHCP 服务软件。仪器设备可简化将来的升级过程，方法是对带有符合要求的 OS 和相应硬件平台的服务版本的升级（程序）进行预打包处理。取决于厂商，这些升级可从单一中心式控制台实施，这种做法免除了物理上安排人员来实施升级的需求。另外，多数厂商支持所部署仪器设备的中心式监测，这使中断或性能降级的提前（proactive）检测成为可能。

当然通常来说，仪器设备要比通用服务器硬件成本高，而且多数都集成 ISC DHCP 服务，对于多数占主导地位的 OS 来说它是可免费获得的，网址是 www. isc. org。在本章中，我们将焦点放在 DHCP 服务的部署策略上，而不考虑是在通用硬件还是在仪器设备平台上的实现方法。

7.2　中心式 DHCP 服务器部署

一般说来，DHCP 服务器的部署归结为如下两方面的折中考虑，一是“比较靠近”客户端的大量服务的广泛部署分配（distribution），一是从各种位置来服务客户端的少量 DHCP 服务器的范围受限的（narrow）部署分配。这种折中考虑的极端情况是，其一在每个子网上有一台 DHCP 服务器，其二是一台或多台 DHCP 服务器处于中心位置，服务机构组织的所有客户端。关键是在客户端和服务器间的 DHCP 服务的可用性和合理的性能之间做出平衡，而同时要保持在服务器和由此导致的可预见未来管理预算约束之内。您所做的部署将极可能处在这两种极端情况之间。

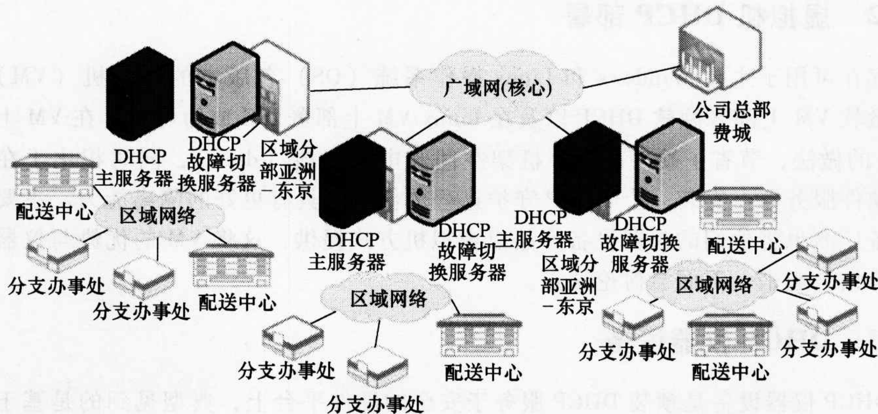


图 7-1　IPAM 全球公司的中心式 DHCP 服务器部署

图 7-1 形象地说明了 IPAM 全球公司的完全中心式部署方法的场景。这个场景重叠在第 3 章图 3-2 的高层网络图之上，其特征是每个区域部署一对 DHCP 服务器，一

台服务器作为主服务器，另一台服务器作为故障切换或备份之用。所有 DHCP 流量必须以隧道方式传输到区域总部站点，对从相应区域到这些站点的鲁棒网络连接形成较高的依赖。这个架构也意味着 DHCP 服务器硬件是足够强大的，从而可满足性能和容量需求。注意，通常情况下，DHCP 主服务器和故障切换服务器应该部署在不同的物理位置上，以便具备抑制灾难的能力。在一个地点的（线路）中断将不会中断一个区域的所有 DHCP 服务。

7.3　分布式 DHCP 服务器部署

在部署连续体（continum）的另一端，去中心化部署方法如图 7-2 所示。在这个图中，一台主 DHCP 服务器位于［附近的］每个分支办事处和配送中心处。这使 DHCP 流量局部化，使用性能不太高的 DHCP 服务器的部署就可满足需求。但是由于 DHCP 故障切换服务器的存在，就仍然具备连通到区域总部的网络能力。这些服务器可作为区域服务器的故障切换服务器，虽然出于负载分担考虑，每个区域要求一台以上的服务器。考虑到负载和冗余能力，您所选中的 DHCP 厂商要具备针对您的网络识别确定可行替代架构的能力。

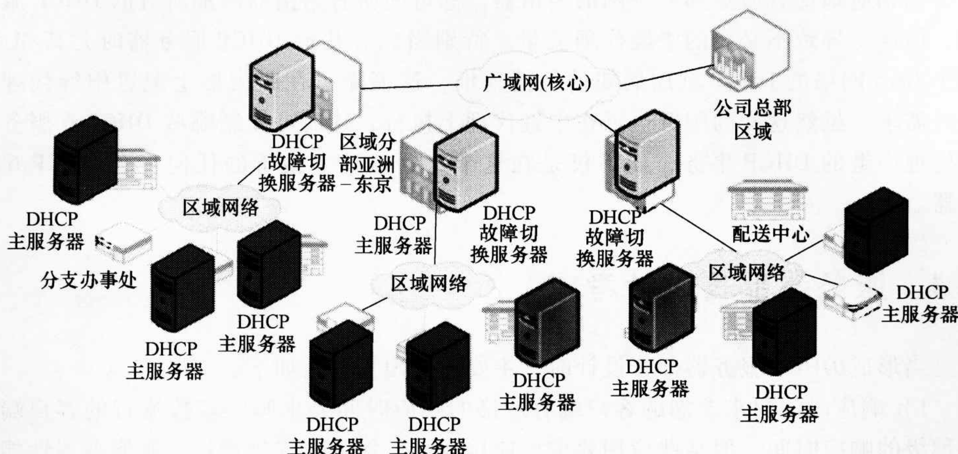


图 7-2　IPAM 全球公司的分布式 DHCP 服务器部署

比较图 7-1 和图 7-2 的两种极端情况，前者要求较少的、但更加强大的 DHCP 服务器和到区域总部地点的坚固的（rock solid）网络连通性。后者则要求更多的 DHCP 服务器，但却是比较中等（modest）性能规格的，以一种网络可达的共享备份方法提供局部化的服务。您可能会存在疑问的是，如果到一个站点的网络链路中断，那么从一台 DHCP 服务器处得到一个 IP 地址有什么益处呢？在没有一条冗余链路的情况下，除了提供到局部网络资源的 IP 接入外，它确实只有有限的价值。但在中心化的架构中，而没有分布式站点时，如果到一个区域总部站点的一条链路出现故障，那么要求新的或刷新地址租期的各客户端将极可能变得一无用处。这和总是常有的事情那样，

必须考虑折中处理方法，通常是中心化与至少部分分布相结合的一种混合法，常常会使总体中断风险最小化。

虽然 ISC DHCP 服务器是单一线程的应用，但对于多数环境而言，其性能通常是足够用的。但是，如果您有数千台 DHCP 客户端尝试在大约同一时间获取地址租赁，则将可能会有一些延迟。如果频繁地出现这种情况，则您可能想考虑部署附加的服务器，并分割为每服务器所服务网络的较精细粒度，以便降低每服务器的负载。同样，通常这不是一个主要担忧的问题，除非您是一个服务提供商，正利用 DHCP 为付过钱的订户初始化如客户端调制解调器的设备，这时会成为一个令您担忧的问题。在从一个临近区域的电源中断恢复之后，各设备将启动恢复，并为得到地址而发出请求，这就淹没了 DHCP 服务器。在这样的环境中，考虑使用一台商用的面向性能的 DHCP 服务器，也许是有道理的。

通过配置您的 DHCP 服务器（在您路由器的中继代理列表内）的 IP 地址，使您的服务器准备好支持 DHCP。在每台路由器内的这些列表，使路由器可终止接收到的 DHCPDISCOVER 报文广播，之后作为单播报文，将之重传到其中继代理列表上的每台配置过的 DHCP 服务器 IP 地址。如果您将网络分隔开，从而使一个给定的 DHCP 服务器服务某些子网的地址池，而其他子网的地址池由另一台 DHCP 服务器服务，要确信您相应地配置服务那些子网的路由器。您可向所有路由器添加所有的 DHCP 服务器，但这将导致不必要的中继代理流量，特别当您有几台 DHCP 服务器时尤其如此。用于 IPv6 网络的 DHCP 利用周知的组播地址，这消除了在路由器上配置中继代理列表的需求，虽然这样的配置也可在中继代理上执行，从而可控制哪些 DHCPv6 服务器来处理中继的 DHCP 事务，而不仅是在这个组播地址上侦听的任何一台 DHCPv6 服务器。

7.4 服务器部署设计考虑

当形成 DHCP 服务器部署设计时，主要考虑因素包括如下。

1）响应时间要求。您的客户端有严格的响应时间要求吗？多数流行的客户端容忍秒级的响应时间，但某些应用要求可能比较高。您的要求越严格，则服务器性能越重要，也许客户端邻近性就越重要。

2）负载要求。您有必须处理的某些负载条件吗？对于利用 DHCP 作为一种顾客端设备（CPE）初始化技术的宽带服务提供商而言，在从驻地电力中断或设备安装或重启中恢复时，可能发生负载尖峰（spike）。对于企业环境，如果几名同事在同一时间或几乎同一时间到达，这样的尖峰会发生在工作日的开始时间，虽然这时许多设备将简单地尝试刷新以前缺省使用的一个 IP 地址。

3）流量预期。您利用短的租赁时间来最小化监管工作吗（这会导致更频繁的刷新尝试）？一般而言，租赁时间（T1 和 T2 时间）越短，则得到租赁和后续租赁刷新尝试之间的间隔就越短。这是来去 DHCP 服务器（可能是多台）的网络上的流量增加，当就前面提到的响应时间要求以及服务器数量和相关联带宽的负载要求，进行设

计时，就必须考虑流量。

4）可用性要求。您的客户端事实上不得不通过DHCP 24×7或是基于“尽力而为”服务，来得到一个IP地址或配置吗？大部分人将回答，高可用性是至关重要的，但随着各设备变得逐渐都是多网络连接的情况出现，只要一个网络的地址指派机制是可用的，则这就是可接受的[㊀]。平均修复时间（MTTR）是满足DHCP服务可用性目标的另一项考虑因素。在本地有一台备用的服务器，会缩短MTTR，同时不得不定购一台替代设备，也将延迟这个过程。

上面的前三个考虑因素与给定租赁分发速率的足够数量服务器部署有关，其目的是满足相应的性能目标。一个好的起点是，在您的网络上每个站点处识别出期望的DHCP客户端数量。这个数应该统计要求DHCP服务的所有设备，包括数据设备、语音设备以及在每个站点要求DHCP服务的所有IP设备。不要忘记统计用户和设备的“峰值”数量，从而使每个人，甚至临时访问的同事，也可得到一个有效的租赁。

在统计DHCP客户端的峰值数量之后，考虑DHCP事务的频率。这将取决于您的租赁时间和客户端租赁释放配置。例如，多数客户端将“记住”一个以前的租赁，当雇员第二天回到办公室工作时，在开机时尝试请求该租赁，虽然情况并不总是这样的。

上面列出的第四项考虑因素与为DHCP客户端提供高可用性的DHCP服务有关。给定提供高可用DHCP服务的一般重要性后，则典型情况下建议为高可用性而实施部署。一旦您依据性能要求，设计部署方案后，就可计划总的或有选择的（selective）高可用性。依据您计划部署的服务器技术，高可用性的实施（implementation）将不仅影响所需要的服务器数量，而且可能影响您的地址空间规划。

ISC DHCP实现和微软DHCP实现利用的是极其不同的方法。ISC服务器利用一个故障切换协议[㊁]，从而对于一个给定的地址池，一台DHCP服务器将作为主服务器，而第二台DHCP服务器将作为备份服务器或故障切换服务器。这个基本配置如图7-3所示。

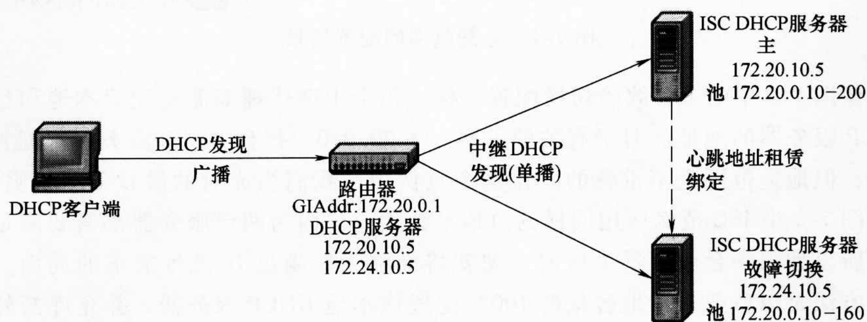


图7-3　ISC DHCP故障切换基本配置

㊀ 当然这个论断假定不同的DHCP服务来服务这些不同的接口，也许情况不是这样。

㊁ ISC实现基于IETF的RFC草案规范，这些草案大部分被搁置。但是，IETF正在尝试重新定义DHCP故障切换协议，ISC计划实现新的版本，同时也支持当前基于RFC草案的实现。

每个中继代理必须配置成将接收到的 DHCP［对于 IPv4］广播报文，单播到主 DHCP 服务器和故障切换 DHCP 服务器，在图 7-3 中是 172. 20. 10. 1 和 172. 24. 10. 1。回顾一下，DHCPv6 中继代理将类似地（likewise）被配置 DHCPv6 服务器地址，或利用一个周知的站点范围组播地址 FF05：：1：3。DHCP 服务器利用一个故障切换协议，从而可使主服务器发送心跳消息和租赁绑定信息给故障切换服务器。故障切换服务器利用用户可设置的参数，来确定主服务器下线了，并开始处理来自中继代理（可能是多台）的单播 DHCP 报文。因此，不管主服务器下线的事实，各客户端仍然能够继续接收 IP 地址和参数指派。一旦恢复，主服务器从故障切换服务器得到当前的租赁数据库，之后再次将其角色设置为主服务器。

微软的方法并不利用一种像 DHCP 故障切换的服务器间协议。相反，通过部署具有互补地址池（不是相同的地址池）的两台 DHCP 服务器，在不用担心重复指派的情况下，任何一台服务器均可处理 DHCP 事务。微软建议使用“80-20 规则”，在一台“本地”服务器上配置地址池的 80%，在一台“远端”服务器上配置地址池的 20%。采取这种方式，假定客户端将首先从本地服务器接收到地址提供，并接受该地址，这种情况下，多数 DHCP 事务将被本地服务器所处理。在图 7-4 中形象地说明了这种配置，其中我们使用 80/20 指导原则将 172. 20. 0. 10 ~ 200 地址池进行了分割。

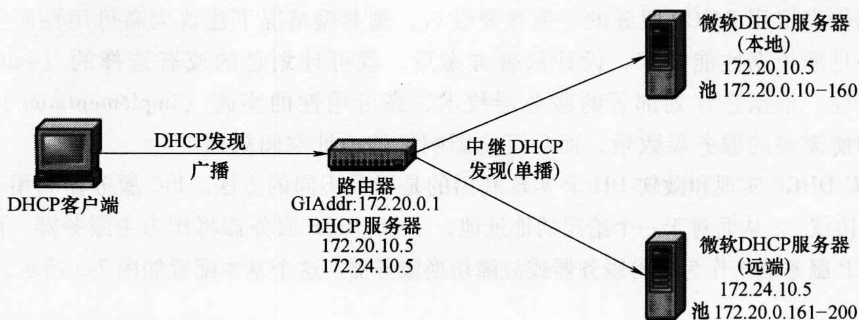


图 7-4　分割范围的配置场景

就像图 7-3 中的 ISC 故障切换配置一样，每个中继代理都需要配置本地和远端微软 DHCP 服务器的地址。对于有关的子网 172. 20. 0. 0，每台 DHCP 服务器都配置一个地址池，但地址范围是不重叠的。在这个范例中，我们为成对的微软 DHCP 服务器，使用与图 7-3 中 ISC 范例所用同样的总地址池大小。因为两台服务器都需要满足容量需求，所以如果一台服务器失效时，您就将面临不能满足 IP 地址需求的局面。另一种替代方法是以所要求本地容量的 100% 配置该本地 DHCP 服务器，并允许额外地址溢出到远端服务器，进行备份。采取这种方式，本地服务器可处理 100% 的容量，且当本地服务器不可用时，远端服务器可协助处理一定比例的那些额外客户端。参见图 7-4，本地服务器可配置有地址池 172. 20. 0. 10 ~ 200，远端服务器配置有 172. 20. 0. 201 ~ 254。这个额外的容量范围可高达 100% 的所需容量，从而在将所需地址空间翻倍的代价下提供 100% 的冗余。虽然是由微软公司普及推广的，但分割地址

范围的方法可用于各厂家的 DHCP 服务器。

从安全角度而言，DHCP 认证的实现没有被广泛地商业化。因此，对于确保 DHCP 事务本身的安全而言，几乎不存在可行的安全措施。对于企业网络而言，这也许不是主要担忧的问题，其中提供 DHCP 用于内部用途，但如果在不知情的情况下在其机器上启动一项 DHCP 服务时，这可能就是有问题的了。也许多数用户不会安装一个 DHCP 服务器，但那些具有（自感知（self-perceived））IT 专业知识的人就可能安装这样的 DHCP 服务器。

对于使用 DHCP 来初始化顾客端（customer）设备的服务提供商网络而言，服务提供商网关或边缘设备的使用，可提供地址指派的 DHCP 客户端有效性的某些保障。将 DHCP 黏结（tying）到配置准备过程，可帮助将一个 DHCP 客户端与一个支付过（paying）的订户标识符相关，从而使服务的被盗窃使用可能性最小化。

DHCP 本身可被当做“安全”网络访问的一种方法，措施是确定一个给定 DHCP 客户端是否满足被网络接纳的可接受准则，这是就服务器的 IP 地址指派而言的。这提供了网络访问控制的一种形式，虽然它并不能防止 IP 地址伪造者的行为。在下一章将讨论访问控制安全的 DHCP 配置。

7.5　在边缘设备上部署 DHCP

多数路由器厂商是将一项 DHCP 服务作为其路由器平台的一个组件提供的。这可能使人们质疑，即为了支持 DHCP 服务，是否需要一个独立的服务器。就多数设计问题而言，答案是“要看情况而定”。有数个站点的小型环境，它有本地服务器服务 100 个左右的不可分的（monolithic）客户端，每个客户端均可由配置路由器提供 DHCP 服务，而得到地址服务。但是，较大型的组织机构或那些要求更高级 DHCP 服务的企业，为了区分语音和数据客户端，以便进行地址和选项参数指派，这时部署分离的（没有集成在路由器内）DHCP 服务器，这些客户端将得到更好的服务。

在一台路由器设备上运行 DHCP 的优势包括如下方面。

1）较低的硬件成本。不需要采购一台服务器或一组服务器。

2）单一用户界面。同一命令行界面可被用来配置路由器和 DHCP 服务器，不需要中继代理配置。

3）“较少的移动部件”。为执行 DHCP 功能，要求的通信链路和服务器都要少一条链路和一台服务器，一般来说，这可增加整体解决方案的可靠性。

在一台路由器上运行 DHCP 的主要劣势如下。

1）选项支持。多数基于路由器的 DHCP 服务器是原始的，它支持地址指派，但在选项支持方面几乎没有。

2）客户端类支持。主要厂商不支持客户端类，而这对于区分将地址/选项指派到不同设备（例如 VoIP 设备和数据设备）来说，是必需的。

3）没有故障切换。如果一台路由器失效，则您在任何情况下都可能丢失连接能

力，但如果出于冗余性考虑，有两台路由器服务一个子网，则必须采用一种分割地址范围的方法，这就增加了管理复杂性。

4）没有中心化的管理。基于路由器的 DHCP 服务，是通过命令行配置的，除非采用一个中心化的工具，否则就 IP 寻址规划而言，就必须手工地配置每台 DHCP 服务器；如果使用多个路由器厂家的产品，则不太可能存在这方面的技术支持。

第 8 章　DHCP 和网络接入安全

安全位于每个 IP 规划人员网络担忧问题列表的首位或接近首位之处。IP 地址管理有关的安全话题也不例外。DHCP 信息方面存在许多安全威胁，在与那些请求的信息通信中也同样存在安全威胁。另外，考虑为接入到网络而分发 IP 地址过程中 DHCP 所扮演的角色，就其固有的功能来说，DHCP 服务自身在提供网络接入控制（NAC）的一个基本层次中扮演了一种关键角色。您将配置 DHCP，使之向任何请求一个地址的设备提供一个 IP 地址吗？抑或您将配置一种更有区分能力（discriminating）的策略？本章将首先深入探究网络接入控制领域，讨论部署审慎精细地址指派策略的通用战略原则。之后我们将讨论 DHCP 信息和通信安全的战术方法。

8.1　网络接入控制㊀

NAC 这个术语是最近几年才流行起来的，但其蕴含的概念是本质上的：在提供这样的接入访问之前，识别是谁正在尝试接入到您的网络。就提供各种层次的接入控制能力而言，存在各种技术。我们将首先开始分析基于 DHCP 的接入控制，必须承认（admittedly）的是，它是 NAC 中较脆弱的方法。之后我们将谈到更广泛可用的（wide-reaching）技术。

8.1.1　采用 DHCP 的区分性地址指派

让我们首先将焦点放在 DHCP 服务以及实现区分性地址指派的一些方法上。存在多数 DHCP 解决方案都提供的几个层次的策略或控制，用于区分是“谁正在请求”一个 IP 地址（通过 DHCP）。第一种方法是简单地依据客户端标识符的一个可用形式（例如请求一个 IP 地址之客户端的 MAC 地址）来过滤请求。回顾一下，MAC 地址是在一条 DHCPv4 报文的客户端硬件地址（chaddr）字段中找到的。DHCPv6 设备标识符由设备唯一 ID（DUID）和身份关联（IA）组成，这两者分别识别每个客户端和接口。

如果 DHCP 服务器有可接受（和/或不可接受）设备标识符的一个列表，则它可如此配置，即向拥有一个可接受标识符的那些客户端提供某个 IP 地址和相关联的参数，不向那些没有一个可接受设备标识符的客户端提供 IP 地址，或仅提供有限功能的 IP 地址。使用“有限功能的 IP 地址”说法，我们指预先配置网络路由基础设施，使之路由的 IP 报文具有仅到某些网络的源 IP 地址（例如仅到因特网，或甚至哪里也去不了）。例如，带有源地址 A 的一条 IP 报文在企业网上是可路由的，而带有源地址 B 的一条 IP 报文仅可路由到因特网。

㊀　本章中的材料依据的是参考文献［11］第 9 章

如我们在第 4 章讨论的，通过在请求一个 IP 地址的设备的客户端类上实施过滤，也可得到这种类型的 IP 地址和配置指派。某些客户端，例如 VoIP 电话，当请求一个 IP 地址时，在 DHCP 报文的厂商类标识符字段提供有关自己的额外信息。也可使用用户类标识符字段。DHCP 服务器可被配置成识别网络上设备的用户类和/或厂商类，以便当设备请求 IP 地址和配置参数时，向 DHCP 服务器提供额外信息。可从某个地址池指派地址，同时可通过标准的或厂商特定的 DHCP 选项向客户端指派附加的配置参数。

通过对请求一个 IP 地址的机器的用户进行认证，则另一种层次的区分 IP 地址指派也是可能的。这项功能可与前面描述的设备标识符和客户端类区别指派法一起使用。例如，如果带有一个未知的或不可接受的设备标识符的一个客户端，尝试得到一个 IP 地址，一种操作选项是完全地拒绝一个地址；另一种操作选项是要求该客户端的用户通过一个安全的访问万维网门户页面进行登陆。

对于您所在网络的合法用户而言，这使其比较容易地捕获新的设备标识符（即有时插入新的接口卡的那些用户）。从 perl 脚本（例如 NetReg（90））到复杂的集成软件解决方案的各种方案都是存在的，可用来将这样的用户定向到一个登录/口令请求网页。之后，对存有合法用户信息的一个数据库的一条简单查询，可允许（或拒绝）客户端访问一个生产（production）IP 地址。这些系统典型地与图 8-1 所示的报文流是一致的[⊖]。

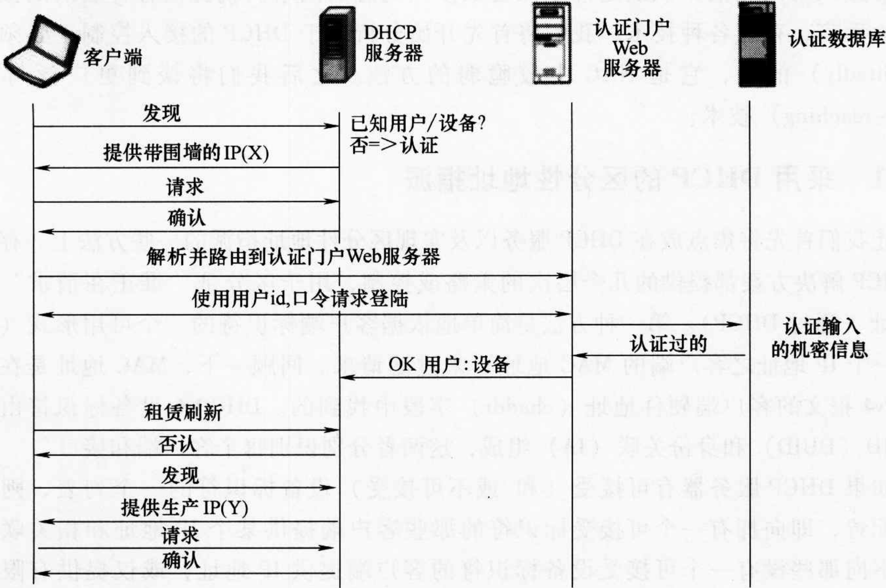


图 8-1 基本 DHCP 受控门户流图[11]

按照这个流图的顺序，流程开始时，是一个设备连接到网络，尝试通过 DHCP 得到一个 IP 地址。DHCP 服务器，利用上面讨论的设备标识符或客户端类-类型过滤法，

⊖ 给出了 DHCPv4 的过程，而 DHCPv6 可利用一个相当的报文流。

确定该设备是否为一个已知的用户设备[⊖]。如果该设备是已知的或以其他方式已经被认证过，那么 DHCP 过程可继续给出一个生产 IP 地址的一条提供报文（Offer），接着后跟一条请求和一条确认。但是，如果该设备是未知的，或要求进行认证，则通过完成 DORA 过程，DHCP 服务器可仍然提供一个 IP 地址；但在这种情形中指派的 IP 地址将是一个受限门户、带围墙的花园，或隔离的 IP 地址。

这些术语指如下事实，即指派给客户端的 IP 地址将仅可被路由到有认证 web 服务器和相关联服务运行的子网或 VLAN。这个隔离的 VLAN 支持 IP 通信，但也仅可到达设备的这个受约束的集合。这将设备封锁起来，使之不能渗透网络的其他部分，直到对应的用户被认证后才解除封锁。路由基础设施必须如下配置，将带有隔离地址池的一个源地址的报文路由到被隔离的 VLAN，同时（或）客户端必须以无类静态路由选项加以配置。因此，如图 8-1 所示的地址 X 是被隔离 VLAN 的一个成员，在该 VLAN 上仅有有限的网络资源可以使用。图 8-2 形象地展示了这种受约束门户配置的一个范例网络拓扑。

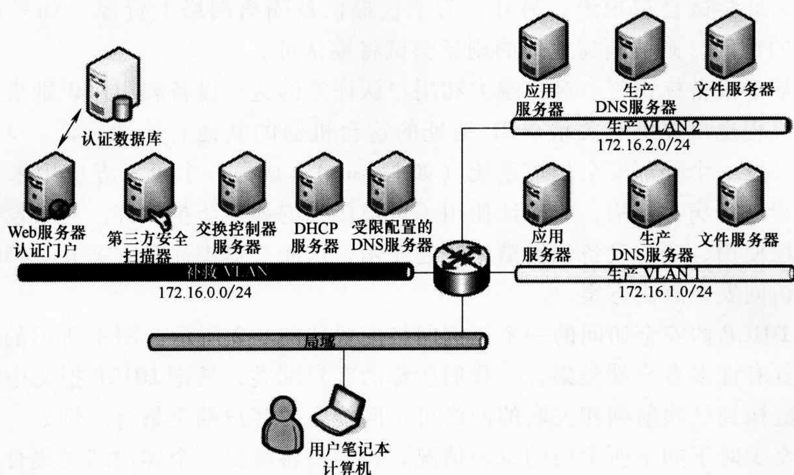


图 8-2 受限的门户网络图[11]

现在，当用户打开一个网络浏览器时，他/她可输入任何网址。在被隔离的 VLAN 上要求有一台受限的配置 DNS 服务器，“受限”的含义指它将任何请求及每条请求都解析到认证网页服务器的 IP 地址。因此，不管输入到网页浏览器的是什么网址，网址都被受限的门户万维网服务器所解析。认证万维网服务器给出登录页面。如果您出外旅行，使用一家旅馆的宽带或无线服务的话，则您可能看到过类似于此的情况。一旦输入被请求的机密信息（对于一个企业环境而言，这典型地将由一个用户 ID 和一个口令组成），则网页会调用一个 CGI 脚本，将输入的机密信息传递到一个后台数据库。这个认证数据库可能是一台 LDAP 服务器、一个 Windows 域控制器、一台 Radius 服务器或其他形式的认证数据库。

依据认证的结果，之后发出请求的设备将认为是授权的或没有授权的，如果被授

⊖ 在一些情形中，甚至已知的用户设备也可请求周期性的再次认证，以此作为一项安全预防措施。

权，则可选地会授予何种类型的授权。相比于一种简单的布尔型“被授权或没有授权”法，授权类提供了更细的粒度，其中不同的被授权用户可被指派一个不同的生产 IP 地址，这顺次可提供到不同网络资源的访问能力。例如，基本层次的用户可被授予访问一个基本的资源集合，而高级层次的用户可被授予访问其他资源（例如 IT 资源）。同样，这要求路由拓扑配置有多个源路由段或 VLAN 段，就与服务层次相关联的地址池而言，将这些网络和相应的路由规划映射到 DHCP 服务器配置。

指派生产 IP 地址的方式，遵循被隔离 IP 地址的超期或刷新拒绝的方式（过程）。一般而言，被隔离 IP 地址租赁时间被配置为一个短的租赁时间（1～5min）。这促使设备快速地尝试刷新。如果该设备仍然处在认证的过程之中，则它的刷新尝试将被确认（ACK），这就延长了租赁时间。一旦成功地完成认证，认证系统就更新 DHCP 服务器，将客户端 MAC 地址添加到“已知的”或“允许的”地址池。之后对被隔离地址的刷新尝试将被否定（NAK），这就激活了一个新的 DORA 过程，从而提供一个“生产”IP 地址（图 8-1 中的地址 Y）。如果设备不能通过认证，则刷新会被否定确认，后续地址尝试会被拒绝；另外，为了仅提供被隔离网络上资源（如果这是所期望的话）的访问，则被隔离地址的刷新尝试将被认可。

除了基于设备标识符、客户端类和用户认证等的这些设备和用户识别措施外，这个通用的流程也可提供有关请求 IP 地址的这台机器的其他有效性验证。DHCP 过程可被用来触发一个外部安全扫描系统（如 Nessus），或另一个第三方应用来对发出请求的客户端进行病毒扫描，或验证使用了可被接受的病毒防护软件。这个设备扫描步骤可被单独使用，或与设备识别措施一起使用，从而可提供通过（采用）DHCP 的一个鲁棒的访问安全解决方案。

基于 DHCP 的安全访问的一个范例网络配置如图 8-2 所示。图中所示的 DHCP 服务器将配置有许多客户端类集合。我们所称的客户端类，是指 DHCP 报文中的匹配准则，将其链接到已映射到相关联的网络可访问能力的客户端类集合。例如，在我们的范例中，至少对于如下所示中的每种情况，我们都将需要一个客户端类集合。

1）受限的门户网络（补救用途（remediation）的 VLAN）。

2）生产网络 1。

3）生产网络 2。

将这些客户端类集合想象为各客户端要被放入的桶（bins），这种做法依据的是将客户端的认证状态与设备的客户端类联系起来。因此，当客户端类成员出现在网络上且用户进行认证时，客户端类成员将被 DHCP 服务器分类，依据的是定义好的客户端类。通常来说，这些客户端类将映射到 DHCP 服务器上的地址池定义，如后面的简单范例 ISC 服务器配置所示[35]。注意，可为每个地址池定义额外的选项，以便向落入每个集合或地址池的客户端提供额外的配置粒度。

```
subnet 172.16.0.0 netmask 255.255.252.0 {
# subnet level options here... /* 下面是子网层次的选项 */
    pool{                          #captive portal pool/* 受限制的门户
                                   地址池 */
```

```
        range 172.16.0.10 172.16.0.254;
        option domain-name-servers 172.16.0.5;  #limited config DNS server/*受限的配置 DNS 服务器*/
        default-lease-time 150;                 #short lease time/*短的租赁时间*/
        allow unknown clients;                  #clients not predefined./*没有预先定义的客户端*/
    }
    pool {                                      #Prod Net 1
        range 172.16.1.10 172.16.1.254;
        option domain-name-servers
        8.1 NETWORK ACCESS CONTROL 131
        172.16.1.5;                             #production DNS server/*生产 DNS 服务器*/
        default-lease-time 14400;               #normal lease time/*正常的租赁时间*/
        deny unknown clients;                   #clients must be predefined./*客户端必须被预先定义*/
        allow members of "net1";                #client class net1 allowed/*客户端类 net1 被允许*/
    }
    pool {                                      #Prod Net 2
        range 172.16.2.10 172.16.2.254;
        option domain-name-servers
        172.16.2.5;                             #production DNS server/*生产 DNS 服务器*/
        default-lease-time 14400;               #normal lease time/*正常的租赁时间*/
        deny unknown clients;                   #clients must be predefined./*客户端必须被预先定义*/
        allow members of "net2";                #client class net2 allowed/*客户端类 net2 被允许*/
    }
}
```

依据认证过程的结果，认证服务器必须能够更新 DHCP 配置，以便将客户端放入合适的容器或类。因此，如果设备被成功地认证可访问生产网络 2，则认证门户需要将特定设备的客户端类值（例如 MAC 地址）添加到生产网络 2 的客户端类组（在上述范例中是“net2”类）。例如这项更新可使用 ISC DHCP 服务器 OMAPI 界面（要求版本 3.1 或以上版本）加以实施。这个客户端类声明可在 DHCP 服务器上定义特定于

类的选项，以便向客户端提供（例如）默认网关、DNS 服务器，还有任何其他选项。

受限的门户 VLAN 可能仅由“未知客户端”组成，它是可采用 ISC DHCP 服务器配置的一个符号指派。受限的门户网络（补救 VLAN）得以部署，包括受限的配置 DNS 服务器，作为认证门户的 web 服务器，可访问一个认证数据库，可选的还有一台安全扫描服务器以及任何其他必备的预访问（preaccess）服务。

为得到高可用性和/或扩展到较大型的网络，可部署一台以上的 DHCP 服务器。这种方法确实使事情有点复杂，原因是在两台服务器上的 DHCP 服务器配置需要做到一致，以便路由未知的客户端或要求到受限门户网络进行认证的客户端。

8.2 其他接入控制方法

您可能认为，对于利用 DHCP 的客户端而言，基于 DHCP 的方法还是不错的；但对于可猜测到子网地址，之后在其机器上人工编码（配置）一个静态 IP 地址来访问网络的那些“狡猾的用户”，该怎么办呢？从一个安全访问的角度而言，这些狡猾的用户毕竟是人们最担忧的那些用户。另外，对于使用 IPv6 无状态自动配置的设备而言，地址指派是不要求 DHCP 交互的。

对于在不依赖基于 DHCP 方法的条件下，要激活设备的检测和相关联的补救动作，有三种基本的替代方法。在下一节我们将讨论领先的网络（networking）厂商 NAC 方法。

1）DHCP LeaseQuery（租赁查询）。

2）层 2 交换机提醒（alerting）。

3）802.1X。

8.2.1 DHCP LeaseQuery

如果在一个子网上的多数地址或所有地址都是依据策略使用 DHCP 加以配置的，即每个 IP 地址应该有一个对应的 DHCP 租赁，那么可使用 LeaseQuery 方法。DHCP LeaseQuery 是一条 DHCP 协议消息，它使一台边缘路由器就一个特定设备或一组设备的租赁状态，来查询 DHCP 服务器。这提供了某种担保，即尝试通过路由器进行通信的一台设备没有伪造一个地址，该地址应该是通过 DHCP 服务器进行指派的。

当路由器从一个特定 MAC 地址在一个层 2 帧内（比如）接收到 IP 流量时，它可向其配置好的 DHCP 服务器（即其角色为中继代理）发出一条 DHCP LeaseQuery 消息。如果一台 DHCP 服务器以前向客户端提供过一次地址租赁，则它将向路由器做出响应，且路由器将打开绿灯，并路由该设备的报文。路由器也可缓存这个信息，从而使 LeaseQuery 速率不会太高。当然，仅当在一个子网上的所有客户端使用 DHCP 时（比如在宽带接入网中的情况），而不是当其他静态编址的设备在该子网上进行通信时，才可实施这种形式的访问控制。

8.2.2 层 2 交换机提醒

另一种方法利用支持（使能）SNMP 的交换机，在其端口之一上遇到一条链路连

通事件时，发出一条SNMP陷阱，并接受端口级别的VLAN配置。这项提醒能力，与SNMP可写配置信息一起，可支持类似网守（gatekeepr-like）的功能，方法是动态地识别尝试访问网络的设备，并配置交换机，从而将端口配置到一个特定的VLAN。为了处理陷阱，将需要一个第三方的系统或产品，在合适的VLAN指派上做出决策，并相应地配置交换机。

让我们看看这是如何工作的。如果我们从开始起考虑连接到一个网络的一台设备的过程，则该设备首先从层1在网络上“启动”（boots up）。因此，首先达到的是物理层/电气连通性；之后数据链路层初始化，在此时发生层2帧同步。接着跟随的是层3，发出一条DHCP报文（比如），或者如果在层3配置了一个静态地址的话，则直接发出IP报文。当数据链路层初始化（在层3之前）时，设备所连接的交换机将被认为是“线路通了”，并发出一条陷阱消息。因为该陷阱是在层3初始化之前发出的，所以这种方案可识别静态编址的和DHCP编址的设备。

陷阱将被发往可识别线路通（link up）状态的一个系统，确认（ascertain）新连接设备的链路层（MAC）地址，之后确定该设备是否要求认证或验证。这个确定过程可通过在系统内的一个MAC地址数据库完成，该系统识别已知的或可接受的MAC地址，并将之从未知的或已知不可接受的MAC地址区分开。系统将这两种或也许更多种MAC地址类别与相应的VLAN指派相关联，之后将其编程在相应交换机的给定端口上。之后连接的设备将被连通到指派的VLAN。您也许看到这类似于我们讨论过的使用客户端类的DHCP场景。在这种情形中，第三方系统使用它的数据库，它并不使用DHCP指派一个IP地址，而是采用SNMP或其他方式配置层2交换机。

对于隔离的或受限的门户访问，VLAN指派将仅导向认证网络。对于传递认证和/或设备验证的那些访问，系统可将MAC地址重新指派到可接受的列表，之后据此配置交换机，以便改变端口VLAN关联关系。取决于认证方法，可能需要也可能不需要客户端软件。对于基于网站的登录/口令，使用认证客户端配置您的每台客户端计算机，也许是不必要的。但是，如果采用Radius或其他挑战/响应认证技术战略措施的话，则客户端软件将是必要的。

8.2.3 802.1X

IEEE 802.1X是支持边缘设备捕获新的访问尝试的一个协议规范，它使用Radius认证和动态交换机端口配置。在宽带因特网前期拨号的日子期间，您可能使用过Radius，在层3使用点到点协议。由IEEE 802.1工作组开发的802.1X将焦点放在层2协议上，和你预料的一样，它是一个层2协议。像在前一节中讨论的基于交换机的认证技术战略一样，这种方法工作在第2层，在设备通过DHCP得到一个层3（IP）地址之前。802.1X是基于标准的，它理论上支持不同厂商的产品用作整体解决方案内的组件。

如图8-3所示，802.1X要求有一个客户端或称为恳求者（supplicant）的代理，它通过一个认证器（例如交换机）的方式与一台认证服务器交互通信。在初步连接到一个网络时，恳求者利用802.1X上的扩展认证协议（EAP）初始化到网络接入设

备的一条连接请求。除了 EAP 报文外，可将交换机配置成默认地阻塞来自未认证端口的所有流量。

在数据链路层上设备所连接到的接入交换机，将 EAP 流量传输到认证（即 Radius）服务器。接下来，Radius 服务器挑战客户端，请其输入一个 ID 和口令。在成功认证后，Radius 服务器与边缘设备通信，支持到关联设备端口的访问。

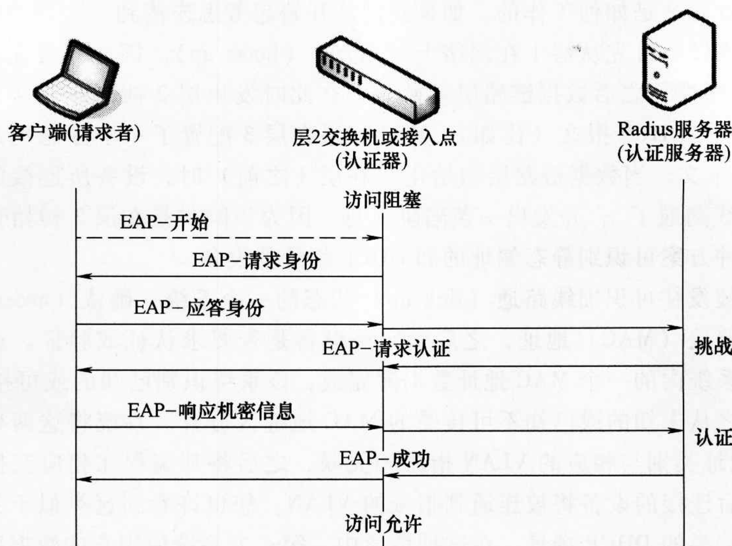


图 8-3 802.1X 认证

8.2.4 Cisco 网络接纳控制

Cisco 的网络接纳控制（NAC）提供法（offering）[91] 主要是基于 802.1X 的。它要求一个 Cisco 信任代理（CTA），可选的是一个 Cisco 安全代理安装在每个端用户设备上（见图 8-4）。信任代理包含一个 Radius 客户端。在初步连接到一个网络时，CTA 利用 802.1X 上的扩展认证协议或 UDP，初始化到该网络的一条连接请求。

网络接入设备，典型情况下是设备在数据链路层所连接的一台交换机，它将 EAP 流量传输到 Cisco 访问控制服务器（ACS），由其提供 Radius 服务。接下来，这个 Radius 组件挑战客户端，请其输入一个 ID 和一个口令。可能触发一个第三方的验证解决方案，来扫描尝试获得接入的设备。在成功认证和验证后，ACS 与边缘设备通信，激活到关联设备端口的访问。

8.2.5 微软网络接入保护

微软引入网络接入保护（NAP）[92]，使管理员能够确保正在访问一个网络的计算机安装了合适的软件，且该软件是一个指定的版本或高于该版本，这项功能在微软 Vista™或 7 客户端以及 Windows Server 2008 实现中得以支持。微软支持一种 API，使其他厂商能够支持 NAP 技术。NAP 主要强调设备的符合性（compliance）和健康状

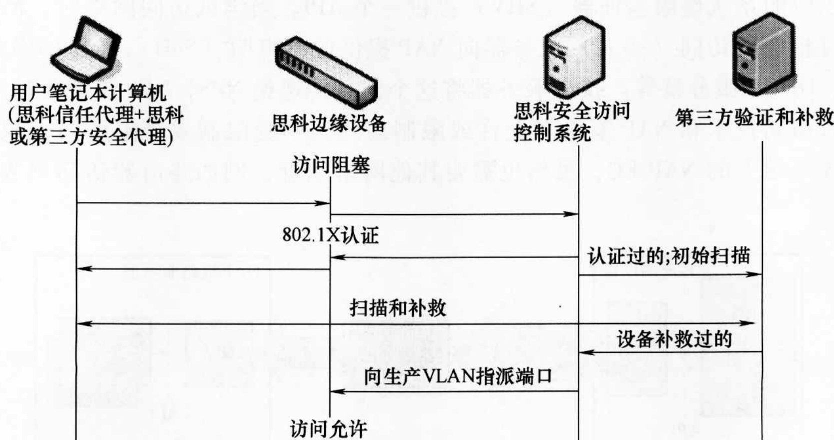


图 8-4　Cisco NAC 基本流

态，而作为一项副产品，才是访问控制。即 NAP 的本旨是当设备访问网络时，使网络管理员能够验证设备与当前软件发行版的符合性，本质上并不预防恶意攻击者的访问。尽管如此，NAP 确实包含了带有其三个主要功能的拒绝访问正在进行中（pending）的健康状态验证能力。

1）健康策略验证。接收到一次网络访问尝试时，得到一台设备的“健康状态”，并将其与管理员定义的“健康策略”比较。如果设备符合指定的健康策略，则允许设备进行不受限制的访问；如果设备是不符合条件的，则可向该设备提供受限制的访问，或仅在监控模式下的全能力（不受限制的）访问。

2）健康策略符合性。在访问尝试时，不符合要求的设备可被可选地自动进行升级。如果处于纯粹监测（monitoring-only）模式，则设备将可使用不受限制的网络访问。在受限的访问模式中，直到取得符合性的能力前，该设备将仅具有有限的网络访问能力。

3）受限的访问。管理员可限制不符合要求的设备的网络可访问性的范围。

微软 Vista 客户端包含一个 NAP 客户端，在访问一个网络的尝试过程中，它与一个 NAP 策略服务器（NPS）（是微软 Windows Server 2008 的组成部分）通信。NPS 加强了符合性策略，并在各种技术上的访问尝试中被查询（consulted）检索，这些技术包括 IPSec、802.1X、VPN、Radius 和 DHCP。IPSec 是策略增强措施的最强壮形式，它由一个健康注册权威（HRA）组成，该权威基于 NPS 所做的符合性验证，向符合要求的 NAP 增强型客户端（EC）发放 X.509 证书。

802.1X 访问流程遵循上面针对 802.1X 描述的流程，添加了 NAP 策略服务器验证设备符合性的步骤。VPN、Radius 和 DHCP 访问组件包括一个 NAP 增强型服务器（ES）和一个 NAP EC，它们在访问网络的尝试中通过对应的技术，就策略符合性进行通信（见图 8-5）。

客户端设备，或 NAP 客户端，包含一个 NAP 代理，以及微软提供的和 API 可访问的系统健康代理（SHA）和 NAP EC，目的是支持另外的应用。类似地，NAP 的特

征是为相应的系统健康验证器（SHV）提供一个 API。当尝试访问网络时，NAP 客户端将通过相应的访问（接入）服务器向 NAP 提供健康声明（SoH）；例如 HRA、VPN 服务器、DHCP 服务器等。接入服务器将这个信息传递给 NPS，NPS 验证策略符合性，基于接入访问技术和 NAP 策略来允许或限制访问。不受限制或受限访问权限被传递到增强客户端上的 NAP EC，虽然也需要其他网络配置，例如路由器访问列表或静态路由。

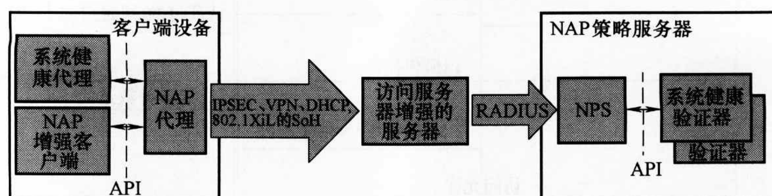


图 8-5 微软 NAP 组件[92]

8.3 使 DHCP 安全

8.3.1 DHCP 威胁

在企业环境内，对 DHCP 的多数威胁是由内部（即组织机构内部）客户端导致的。外部客户端不应该到达（即连通）DHCP 服务器，方法是，简单地不将 DHCP 服务器部署在外部子网，也不中继来自外部源的 DHCP 报文。对于使用 DHCP 初始化订户设备（不管是蜂窝电话（手机）、线缆或光纤路由器等）的服务提供商而言，对 DHCP 服务的威胁，依据定义是从外部到达网络的。简而言之，使用 DHCP 的所有组织机构都是脆弱的。脆弱的程度以及被攻破的影响，应该以使 DHCP 安全的形式，以便最小化这种影响，来驱动响应行为。接下来，我们将审视一下攻击的主要形式。

像所有网络服务一样，对于拒绝服务（DOS）攻击，DHCP 是脆弱的。当一个攻击者以对服务器而言太多的请求，使其不能处理，而洪泛攻击一台给定服务器时，该服务器将用其所有的计算周期，尝试处理洪泛请求，而不能处理合法的客户端请求；因此，这些合法的客户端就不能被服务，对它们而言服务是不可用的。

另一种类型的攻击，涉及一名无赖（rogue）客户端尝试得到一个有效的 IP 地址和配置，以便可访问网络。这可能是恶意的（例如宽带服务被盗窃使用），或简单的偶然性的（例如一名拜访者将网线头插入会议室的墙上插座）。

第三种形式的攻击，特征是，一台无赖 DHCP 服务器对来自客户端的租赁请求做出响应，提供不正确的 IP 地址和/或选项参数信息。这种“中间人”型的攻击会尝试在客户端上设置不正确的配置参数，例如所用的默认网关或 DNS 服务器地址（可能是多个地址）。注意对于 IPv4，一般而言，仅当服务器和客户端位于同一子网上时，一台无赖 DHCP 服务器攻击才是可行的；假定中继代理将被配置成去中继 DHCP 报文到经过授权的 DHCP 服务器。一台远端的无赖 DHCPv6 服务器可能通过 DHCP 组播地

址是可到达的。

客户端可从合法的 DHCP 服务器（可能是多台）和无赖服务器，接收 DHCPOFFER。许多客户端将选择包括其请求参数的第一个提供（offer）报文。如果无赖服务器和客户端处在同一子网上，而合法服务器不在这个子网上时，那么可能情况是，无赖服务器也许能够指派规定客户端的 IP 配置。

8.3.2 DHCP 威胁缓解措施

针对 DOS 攻击的防护，应该在超出 DHCP 的一个更广泛的上下文中进行实现。在一个组织机构内的其他潜在被攻击目标，包括 DNS 服务器或 web 服务器，隐含着要考虑一种基于网关的或报文过滤的方法，以便以一种通用的解决方案来保护所有的服务器。典型情况下，这样一种解决方案涉及报文过滤和正在进行的未完成报文数量的阈值限制，但就 DHCP 而言要谨慎从事，这里的原因是多数客户端的事务都是以漏斗方式（funnel）通过 DHCP 中继代理的，它将来自一个给定源地址集合的报文集中处理。

通过不正当地从 DHCP 得到一个 IP 地址，从而得到 IP 网络访问权限的未知客户端，这种情况会构成威胁，缓解这种未知客户端威胁的步骤要求识别客户端，依据是我们在本章开始讨论的各种访问控制技术。

无赖的 DHCP 服务器是难以检测的，由于对于与无赖服务器处于相同子网上的各客户端更是如此。但 ISC 和微软的实现都提供了缓解无赖服务器危害的方法。ISC 使用 authoritative（权威的）命令（directive），该命令配置服务器，使其当一条客户端请求一个地址租赁时（该服务器是这个地址的权威服务器），但该服务器却没有记录，这时服务器发出一条 DHCPNAK。微软的实现则要求 DHCP 服务器在活跃目录（Active Directory）内得到授权；因此，当一台 Windows DHCP 服务器启动时，在处理 DHCP 报文之前，它要在活跃目录中验证它的授权。

8.3.3 DHCP 认证

IETF 在 RFC 3118[47] 中定义了 DHCP 认证，将之定义为通过使用共享令牌或密钥（key），提供 DHCP 报文发送者和接收者验证的一种机制。简单地说，一个令牌就是一个固定数值，它被插入到 DHCP 认证选项字段。报文接收者检查令牌，如果令牌与它所配置的令牌相匹配，则接受该报文；否则，它丢弃报文。这种方法提供了弱的端点认证，但不提供消息验证。使用共享密钥的方法，可提供带有消息验证的更强壮的端点认证。但是，共享密钥必须被配置在每个客户端上，且每个客户端的密钥要配置在每台 DHCP 服务器上（正是通过它们，客户端得到地址租赁的）。DHCP 认证规范没有定义密钥分发的机制。例如，移动客户端将需要可能与其交互通信的每台 DHCP 服务器配置相应的令牌，反之亦然。

下面是 DHCP 认证如何工作的过程。客户端产生其 DHCPDISCOVER 报文的一个 HMAC-MD5 散列，并使用共享密钥对其签名。得到的摘要被放置在 DHCP 认证选项之中，并在 DHCPDISCOVER 报文内传输到服务器。出于散列计算的目的，DHCP 认

证选项的散列部分必须设置为0。之后DHCP服务器将利用与客户端相关联的共享密钥（由DHCP认证选项的秘密ID字段加以识别），计算所接收到消息的一个散列值。出于散列计算的目的，服务器将散列值、跳数和GIAddr字段清零。如果计算得到的散列值与原始DHCP认证选项中传输过来的散列值相匹配，则认为客户端和报文内容是被认证过的。当DHCP服务器准备它的DHCPOFFER和发往客户端的未来报文时，它使用相同的共享密钥来计算其DHCP认证选项的散列值。

存在非常稀少的DHCP认证实现。对于人们可感知到的认证优势而言，因为密钥管理和由于散列计算导致的处理时延这两方面的挑战，认证被认为是一项太繁重的负担，而无法为人们所承受。那么，典型情况下，DHCP服务的安全责任就落在DHCP服务器管理员的身上，由他们监测服务器，当出现威胁时，对威胁做出响应。

第Ⅲ部分　域名系统（DNS）

DNS 提供了一项自动化的查找设施，目的是方便人们使用 IP 网络。虽然 DNS 提供的最常见的查找功能是将名字解析为 IP 地址，但我们在第 9 章中讨论 DNS 协议基本知识之后，正如我们将在第 10 章看到的，DNS 可支持种类很广的应用。我们将在第Ⅲ部分的其他章节中讨论部署和安全。

第 9 章　DNS 协议

9.1　DNS 综述——域和解析[⊖]

DNS 是 IPAM 的第三个基础和 IP 通信的一个基本单元。DNS 提供了如下方面的各种方法，它提供了 IP 应用的可用性改进，隔离端用户使其不必将 IP 地址直接输入到像网页浏览器的应用之中。当然，要在一个 IP 网络上通信，一台 IP 设备需要将 IP 报文发送到预期的目的地 IP 设备；正如我们已经看到的，IP 报文首部要求源地址和目的 IP 地址。DNS 提供了从一个用户输入的命名目的地信息（例如网站地址）转换到其 IP 地址的手段。

作为一项网络服务，DNS 已经从简单的主机名称到 IP 地址查找设施，演化发展到支持非常复杂的“查找”应用，支持语音、数据、多媒体和安全应用。对于这样的查找功能，已经证明 DNS 是极具扩展性和可靠性的。在开始介绍这个信息是如何组织的之后，我们将讨论这个查找过程是如何工作的。

9.1.1　域层次结构

全球域名系统实际上是一个分布式的层次化数据库。一个域名中的每个“点”指明层次结构中各层之间的一个边界，在点之间的每个名字表示一个标签（label）。层次结构的顶部，“.”或根域提供了到顶层域的索引，例如 .com、.net、.us、.uk，这些顶层域接下来索引相应的子域。每个这样的顶层域（Top Level Domain，TLD）是根域的一个子域。每个 TLD 也有几个子域，例如 ipamwordwide. com 作为 com 域的 ipamworldwide 域。这些子域可能有子域等。

当我们从右到左在点之间读取信息时，可识别我们正在寻找的主机的一条唯一

⊖　本章的开始几节依据的是参考文献［11］的第 4 章。

路径。最左侧点左边的文本[⊖]通常是主机名，它位于域名其他部分所指明的域的内部。一个完全合格的域名（FQDN）指在全球 DNS 数据层次结构内对节点或主机而言，这个唯一的完整［绝对］的路径名。图 9-1 形象地说明一个完全合格的域名映射到树状结构的 DNS 数据库。注意在 . com. 之后的尾部‘点’号，它显式地表明域名内部的根域，这使之成为完全合格的。记住，如果没有这个显式的 FQDN 尾部‘点’表示的话，则一个给定的域名可被二义性地解释为是完全合格的，或相对于“当前”域的。这当然是合法的和较容易的速记法，但一定要小心潜在的二义性。

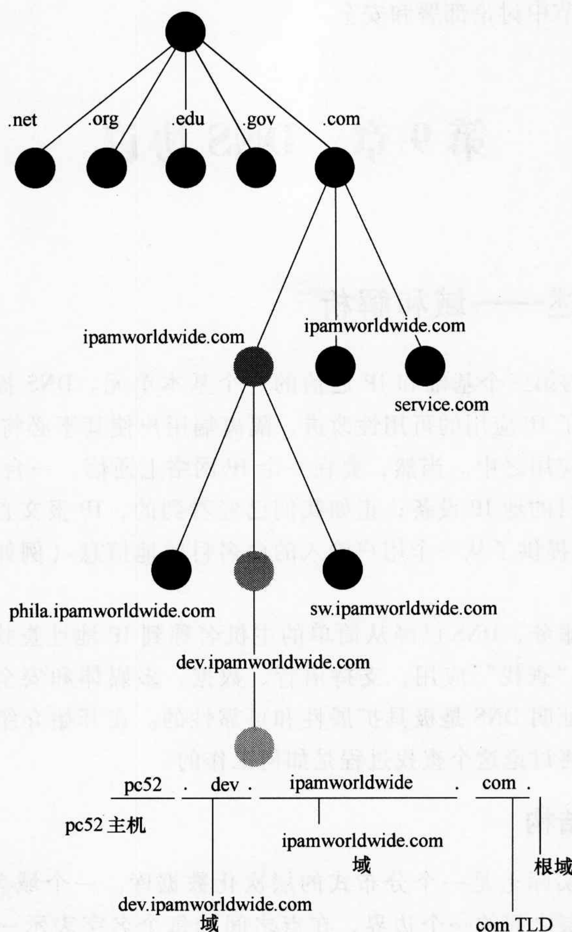


图 9-1 映射到一个完全合格域名的域树[11]

9.2 名字解析

为了形象地说明域信息是如何组织的以及一台 DNS 服务器如何利用这个层次化

⊖ 一些环境允许主机名内出现“点”，这相对而言是不常见的，虽然是被允许的。

数据结构的，让我们看一个名字解析范例。依据图 9-1 的范例，让我们假定我希望连接一个名为 pc52 的设备。因此我输入主机域名 pc52. dev. ipamworldwide. com，作为我期望的目的地。我将这个域名键入其中的应用（例如电子邮件客户端和网页浏览器）利用套接字㊀应用编程接口（API），与 TCP/IP 栈内称为（解析器）resolver 的一部分代码通信。在这个实例中，解析器的工作是将我键入的 web 服务器名转换为一个 IP 地址，使用这个 IP 地址可发起 IP 通信。

解析器向我的本地 DNS 服务器发出这个主机名的一个查询，请求该服务器提供一个答案。这台本地 DNS 服务器的 IP 地址是采用手工配置㊁的，或使用域名服务器选项（DHCP 中的选项 6 和 DHCPv6 中的选项 23）通过 DHCP 配置的。之后这台 DNS 服务器将尝试回答查询，方法是以指定顺序在后根（following）的区域中查找，如图 9-2 所示。

我们经常称解析器向其发出查询的这台 DNS 服务器为一台递归服务器。“递归”意味着解析器希望，如果 DNS 服务器自己不知道答案，也要尝试找到其查询的答案。从解析器的角度看，它发出一条查询，并期待一个答案。从递归 DNS 服务器的角度看，它尝试为解析器定位找到答案。递归服务器是解析器进入全球域名系统的“门户”。递归服务器直接从客户端解析器接收递归查询，并实施如下列出的步骤，代表解析器得到查询的答案。

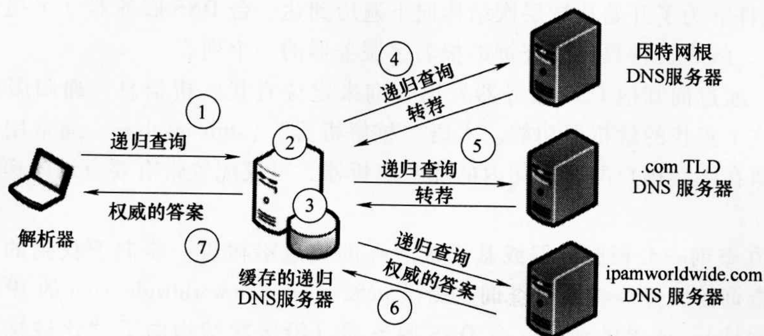


图 9-2 名字解析中的递归查询和重复查询[11]

（1）解析器向递归 DNS 服务器发起一条查询。依据通过手工输入或 DHCP 的配置信息，解析器知道要查询哪台 DNS 服务器。

（2）被查询服务器将首先搜索其所配置的数据文件。即在典型情况下，DNS 服务器配置有配置和资源记录信息，它是这些信息的权威服务器。在典型情况下，这个信息是使用文本文件、一个 Windows 界面或一个 IPAM 系统配置的。例如，贵公司的 DNS 服务器可能配置有贵公司 IP 设备的解析信息。如此，这就是权威信息。如果找到答案，则将其返回解析器，过程终止。

（3）如果被查询的服务器不是被查询域的权威，它将访问它的缓存，以便确定

㊀ 这个 API 是位于从应用到协议栈的 TCP/IP 层的。gethostbyname sockets/Winsock 调用发起这个特定的过程。

㊁ 我们将在本章稍后部分回顾如何进行这种手工配置。

它最近是否在以前的解析任务过程中，从另一台 DNS 服务器接收到过相同的或类似的查询。如果 pc52. dev. ipamworldwide. com. 的答案还在缓存[1]之中，则 DNS 服务器将以这个非权威的信息返回给解析器，过程终止。这不是一个权威答案的事实，一般来说，是没有什么不良后果的，但服务器在其响应中，将这个事实提示给解析器。

(4) 如果被查询的 DNS 服务器不能在缓存中定位找到被查询的信息，那么它将尝试通过另一台有该信息的 DNS 服务器来定位查找该信息。用来执行这种“逐步增强”功能，存在三种方法。

1) 如果在步骤 (3) 中，索引的缓存信息指明查询的一个部分答案，则它将尝试联系那个信息的源，来定位最终的源和答案。例如，对另一台 DNS 服务器——服务器 A 以前的一条查询，可能指明那台 DNS 服务器 A 是 ipamworldwide. com 域的权威。那么初始时被查询的 DNS 服务器会查询 DNS 服务器 A，查询得到 pc52. dev. ipamworldwide. com 的解析。

2) 如果缓存没有提供相关的信息，且被查询的递归服务器被配置为可转发状态，则该服务器将依据其配置或区域 (zone) 文件的配置转发该查询。在后面我们将讲解细节内容。

3) 如果在缓存中没有找到信息，该服务器不能识别一台转荐 (referral) 服务器，或转发没有提供一个响应[2]或没有配置成可转发，则 DNS 服务器将访问它的线索 (hints) 文件。为了开始从域层次结构向下遍历到达一台 DNS 服务器 (可提供查询的一个答案)，线索文件提供要查询的根名字服务器的一个列表。

注意，通过向其他 DNS 服务器发出查询来定位查找解析信息，递归服务器本身实施执行这个查找的解析器功能。术语“桩解析器” (stub resolver) 通常用来识别解析器，就像在终端用户的客户端内的那些解析器，可仅配置带有要查询的递归名字服务器 (NS)。

(5) 在查询一台根服务器或基于被缓存的信息沿树进一步向下找到的一台服务器时，被查询的服务器将解析查询，提供 pc52. dev. ipamworldwide. com 的 IP 地址 (可能有多个地址)，或提供到另一台 DNS 服务器 (沿层次结构向下“比较接近”要查找的完全合格的域名) 的一条转荐信息。例如，在查询一台根服务器时，可以有保证地说，您将不会得到 pc52. dev. ipamworldwide. com 的一个直接解析答案。但是，根名字服务器将正在查询的 DNS 服务器指向负责 com. 的权威名字服务器。根服务器是“仅被委派”的服务器，不直接解析查询，仅对被查询的 TLD 返回被委派的名字服务器信息。

(6) 直到查询被回答之前，递归服务器依据沿域树 (domain tree) 向下得到的应答，迭代重复[3]附加的查询。继续我们的范例，在查询 com. 的权威名字服务器时，接收到的答案将是 ipamworldwide. com. 的权威名字服务器的一个索引，如此沿树向下

[1] 缓存表项是临时的，DNS 服务器依据用户配置信息以及一个资源记录的通告寿命，对其进行清除。

[2] 如果配置仅转发 (forward only) 选项，如果被转发的查询返回的是没有结果，则不做解析尝试；如果配置首先转发 (forward first) 选项，则接着发生本段中列出的过程，并逐步到达一台根服务器。

[3] 这些“点到点”查询被称作迭代查询。

遍历。最终，应该可定位得到与该查询有关的区域的权威 DNS 服务器。权威 DNS 服务器将读取所查询类型的一个资源记录的相应区域信息。该服务器将资源记录（可能有多条记录）传递给发出查询请求（递归）的 DNS 服务器。

（7）当接收到答案时，递归 DNS 服务器将向解析器提供该答案，同时更新其缓存，过程终止。如果没有找到一个答案，递归服务器也将缓存这个“负面的”信息，用于对类似查询做出响应。

总之，解析过程包括①寻找带有权威信息的一台名字服务器，以便解析存在疑问的查询；②向那台服务器查询期望的信息。在我们的范例中，期望的信息是对应于域名 pc52. ipamworldwide. com 的 IP 地址。将被查询域名映射到一个 IP 地址的这个“翻译”信息，以一条资源记录的形式被存储在 DNS 服务器之中。为不同类型的查询定义了不同类型的资源记录。每条资源记录包含一个“主键”（key）或查找值和一个对应的解析或答案值。在一些情形中，一个给定类型的一个给定查找值在 DNS 服务器配置中可能有多个表项。在这种情形中，权威 DNS 服务器将返回资源记录（或 RRSet）的整个集合，它匹配被查询的值（名字）、类和类型。在下一章我们将详细讨论资源记录。

底部一行是，DNS 服务器在其相应域信息为权威的域树的所有层次上进行配置，同时还有沿域树向下到哪里去索引查询器。在许多情形中，在不同层次的这些服务器是被不同组织机构管理的。不是在域树中的每个层次或节点都需要不同的 DNS 服务器集合，原因是一个组织机构会在一个 DNS 服务器的同一集合内服务多个域层次。

虽然域树的上三层典型地使用不同管理权威之下的三个 DNS 服务器集合，但在 DNS 服务器单一集合上支持一个组织机构内部的多个层次或域却是一项部署决策的事情。这个决策主要根据是否要求（或期望）管理委托进行授权。例如，ipamworldwide. com 域的 DNS 管理员会期望保留 dev. ipamworldwide. com 域的管理控制权限，但将 eng. ipamworldwide. com 委托给一组不同的管理员和名字服务器。这引出我们在下面就区域和域之间的不同进行讨论。

9.3 区域和域

术语“区域”（zone）是就域（domain）层次结构而言，区分管理控制的层次的。在我们的范例中，ipamworldwide. com 区域包含 ipamworldwide. com 和 dev. ipamworldwide. com 域的权威，而 eng. ipamworldwide. com 区域保留 eng. ipamworldwide. com 域的权威，如图 9-3 所示。

通过委托 eng. ipamworldwide. com 的权威，ipamworldwide. com 的 DNS 管理员们同意将 eng. ipamworldwide. com（以及域树中 eng. ipamworldwide. com 之下的任何子域）的所有解析传递给运行 eng. ipamworldwide. com 区域的员工所管理的 DNS 服务器。这些 eng. ipamworldwide. com 管理员们可自治地管理他们的域和资源记录以及任何子域；他们仅需要通知父域管理员们（ipamworldwide. com 的）当解析器或其他 DNS 服务器尝试沿域树向下遍历搜索解析时，将他们接收到的查询定向到哪里。

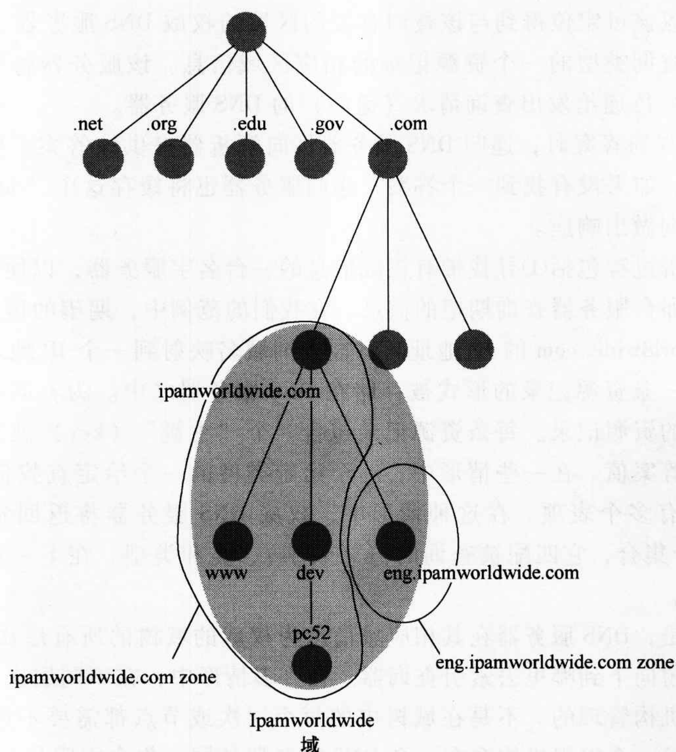


图 9-3　作为被委托域的区域[11]

因此，ipamworldwide. com 区域的管理员们必须为 ipamworldwide. com 区域配置所有的资源记录和配置属性，包括 ipamworldwide. com 区域内部的各子域，例如 dev. ipamworldwide. com 域。同时，ipamworldwide. com 管理员们必须提供到任何子区域（例如 eng. ipamworldwide. com）的一条委托连接关系。这条委托连接关系是以如下方式支持的，在将名字服务器（NS）资源记录输入到 ipamworldwide. com 区域文件内，指明哪台名字服务器是 eng. ipamworldwide. com 被委托区域的权威。通过沿域树向下索引解析器或其他名字服务器，这些 NS 记录提供了到被委托子区域的连续性。称作粘接记录的对应 A 或 AAAA 记录也通常被定义将被解析 NS 主机域名粘接到一个 IP 地址，从而支持进一步查询的直接寻址。

图 9-3 中的阴影表明，ipamworldwide. com 域包含 ipamworldwide. com 节点及其所有的子节点，突出显示了 ipamworldwide. com 这个层次的责任关系及其“下”的情况。ipamworldwide. com DNS 管理员们负责维护 ipamworldwide. com 区域的所有 DNS 配置信息以及到服务被委托子区域 DNS 服务器的转荐信息。因此，当世界各地的其他 DNS 服务器（代表其客户端）在尝试解析终结于 ipamworldwide. com 中的任何名字时，它们的查询将要求遍历 ipamworldwide. com DNS 服务器，也许还有其他 DNS 服务器，例如服务 eng. ipamworldwide. com 区域的那些服务器。

将名字空间委托的过程支持 DNS 配置的自治，同时通过全球 DNS 数据库内部的 NS 记录转荐提供连接关系。正如你所想象的，如果这些 NS 记录索引的名字服务器不

可用的话，则域树将在那个点是断开的，使域树中在那个点或之下的名字不可解析。如果 eng. ipamworldwide. com DNS 服务器下线，则 eng. ipamworldwide. com 及其子节点的权威解析将会失效。这形象地说明，出于冗余性考虑，要求每个区域必须至少有两台权威 DNS 服务器。

因此 ipamworldwide. com 区域的管理员将以 ipamworldwide. com 和 dev. ipamworldwide. com 域的配置和解析信息来配置他们的 DNS 服务器；他们也将以服务被委托区域或子区域的 DNS 服务器的名字和地址配置他们的服务器。他们不需要知道这些被委托区域的任何知识；仅仅知道要联系谁，从而一个转荐可被发送到发出查询请求的递归 DNS 服务器。

DNS 服务器配置信息由服务器配置参数和所有区域（该服务器是这些区域的权威）的声明组成。这个信息可在作为一个给定区域集的权威的每台服务器上加以定义。资源记录的添加、改变和删除、每个区域配置文件内的离散（discrete）解析信息，可在一台主（master）服务器上输入一次，或更准确地说，被配置为相应区域主服务器的服务器上输入一次。类似地也是这个信息的权威的其他服务器可被配置为从属的（slaves）或辅助的（secondaries）服务器，通过区域传递的过程，它们得到区域更新。区域传递使一台从属服务器从主服务器得到权威区域信息的最新拷贝。集成微软活跃目录的 DNS 服务器，出于与这个标准过程兼容的考虑，支持区域传递，但也支持 DNS 数据复制，方法是使用本地的活跃目录复制进程。

9.3.1 区域信息的传播

考虑到 DNS 服务在解析权威信息并维护域树连接关系中的关键地位，则 DNS 服务器冗余就是必需的。不同的 DNS 服务器厂商采取不同的冗余方法。当 DNS 信息被集成到活跃目录（这是 Windows Server 产品的架构基础）时，微软将 DNS 信息在一组域控制器间复制。ISC BIND 实现通过一个轮辐型模型（hub-and-spoke）支持 DNS 信息复制。配置改变是依据上述方法输入到一台主 DNS 服务器的。冗余 DNS 服务器被配置为从属的或辅助角色，它们通过区域传递的过程得到区域更新。区域传递使一台从属服务器能够从主服务器得到其权威区域信息的最新拷贝。出于与这个标准过程的兼容性考虑，微软的活跃目录集成的 DNS 服务器也支持区域传递。

区域文件的版本由一个区域序列号跟踪记录，该序列号在每次一个改变施加到区域时，就会发生改变。将从属服务器配置为周期性地检查设置在主服务器上的区域序列号；如果为区域定义的序列号大于其自己的值，则它会做出结论，即它有超期的信息，并将发起一次区域传递。另外，该区域的主服务器可被配置成通知其从属服务器，发生了一次改变，促使从属服务器立刻检查序列号，并实施一次区域传递，以便比等待正常的周期性更新检查更快速地得到更新。

区域传递可由称作一次绝对区域传递（AXFR）的整个区域配置文件组成，或仅由称作一个增量区域传递（IXFR）的增量更新组成。在区域信息是相对静态的并从单一源更新的情形（例如一名管理员）中，依据需要而采用 AXFR 的序列号检查法运行良好。这些所谓的静态区域相比其对应物：动态区域，对于管理员而言，是要简

单得多的。正如名字所隐含的，动态区域从 DHCP 服务器（比如）接收动态更新，以新指派的 IP 地址和相应的域名更新 DNS。动态区域的更新方法可利用 IXFR 机制，在主服务器和多个从属服务器间维持同步。

对于 BIND 9，在每台服务器上的日志文件，可提供跟踪区域信息动态更新的一种高效方法。这些日志文件是相应区域文件的临时附属物，直到服务器将这些日志表项写入到区域文件并重新载入区域信息之前，支持动态更新的跟踪记录。许多服务器实现将区域文件信息载入内存，还有为了快速解析，它也将增量区域更新载入内存。我们将在本章后面详细讨论服务器和区域信息，但首先让我们考虑不同种类的域树结构。

9.3.2 反向域

直到此时，我们介绍了常见的名字到 IP 地址的解析过程，它为一个名字解析定位一台权威的 DNS 服务器，该服务器之后对查询做出权威响应。查询的另一种普遍形式是 IP 地址到名字的解析。解析的这种“反向（Reverse）”形式被普遍用作一种安全检查，是当建立虚拟专用网（VPN）连接或通用的 IP 地址到主机名查找时才使用的。给定一个 IP 地址，一台 DNS 服务器如何遍历域树来找到一个主机域名呢？在称作地址和路由参数区（ARPA）的域内为基于 IP 地址的域树定义了特殊的顶层域：为 IPv4 地址到名字解析定义的 in-addr. arpa，为 IPv6 地址到名字解析定义的 ip6. arpa⊖。

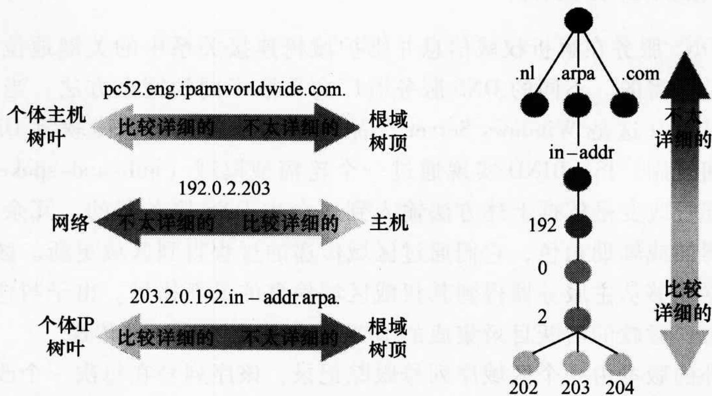


图 9-4 IP 地址反向域树映射[11]

在一个域树内组织 IP 地址的唯一缺陷来自于映射一个 IP 地址，它从左到右读，即从不太详细的（网络）到比较详细（IP 主机）信息，而读取一个域名是从左到右，先是比较具体的（特定主机、域）到不太具体的（根）。因此，将 IP 地址反向，以便支持域层次结构内部的表示，从左到右读取，从比较具体的信息到不太具体的信息。在图 9-4 中对此进行了形象地图示。

⊖ 从技术角度看，这些都是. arpa. TLD 的子区域。

您可能注意到，点分十进制表示的映射方法，支持反向域到基于字节边界的网络分配的映射。例如，如果我们被分配一个 C 类网络 192. 0. 2. 0/24 作为我们的公共空间，就可容易㊀地将如上所示的 in-addr. arpa. 域的叶子映射到个体主机。就像主机名的解析一样，对 in-addr. arpa. 域树的遍历遵循对地址到名字查询的权威解析的一个类似过程。指针（PoinTeR，PTR）资源记录，提供从地址到主机的一个映射，我们将在下一章中讨论。

但如果我们被分配的子网不在字节边界，情况会如何呢？例如，如果我们被分配一个/23 子网，而不是一个/24 子网，则网络地址可被表示为 192. 0. 2. 0/23。这个/23 子网实际上由两个/24 网络组成：192. 0. 2. 0/24 和 192. 0. 3. 0/24。对应于这些字节归一化（normalized）的网络地址的两个反向域 2. 0. 192. in-addr. arpa 和 3. 0. 192. in-addr. arpa，将需要配置在 DNS 内部，从而允许这个/23 网络内主机的反向查找。

如果所分配的子网要比一个 C 类网络小，则需要一种更复杂的表示和区域文件配置。举个例子，假定我们为一个远端办事处分配一个子网 192. 0. 2. 0/25。如果我们尝试将对应的反向域表示为 2. 0. 192. in-addr. arpa，这将包括 192. 0. 2. 0/24 网络的期望的一半网络，但也包括了该网络的“另一半”网络，即 192. 2. 128. 25 网络。但这另一半网络可能分配给了一个不同的组织机构，它有其自己的 DNS 权威（授权）(authority)。在那种情形中，因为有类的（classful）反向区域分割到了两个权威机构，那么谁将管理这个有类的（classful）反向区域呢？解决方案是指明，将第 4 字节的那部分对应到反向域名字所应用的子网。

RFC 2317[93] 规范了在 in-addr. arpa 区域名内使用 CIDR 表示法。因此，直接将各点之间所分配子网的号码反向，我们得到如下：对于网络 192. 0. 2. 0/25，对应的反向域是 0/25. 2. 0. 192. in-addr. arpa㊁。这个 C 类网络的“另一半”将是 128/25. 2. 0. 192. in-addr. arpa。较小规格的子网将遵循一种类似的表示法，它使用网络地址的第四个字节，接着是/<网络规格>，接着是 IP 地址反向的其他三个字节，之后附接上 in-addr. arpa。

但当一个解析器发出一条查询时，它将以 185. 2. 0. 192. in-addr. arpa. 的形式查询一个特定的地址（PTR 记录），那么在 128/25. 2. 0. 192. in-addr. arpa 的情形中，我们如何将这个查询映射到合适的区域文件呢？解决方法是（call for），使用父区域(2. 0. 192. in-addr. arpa. ）中的规范名（CNAME）记录，有选择地指向合适的被委托区域，每个被委托区域可能由独立的 DNS 管理员所管理。一个 CNAME 记录用作一条给定记录的一个别名，之后引导查询器查询该别名。在这种情形中，需要针对每个个体 IP 地址产生一条 CNAME 记录，并映射到一个对应的 RFC 2317 风格的反向区域，

㊀ 当然“容易”是一个相对的术语，但一旦您熟悉了反向域，则至少这样的有类网络是容易可视化为反向域的。

㊁ 虽然 RFC 2317 规范了在这些域名内使用斜线（/），但许多 DNS 管理员以短线（-）替换之，目的是将区域名与文件名关联起来，文件名是不能包含斜线的。因此，我们可将这个区域表示为 0-25. 2. 0. 192. in-addr. arpa，它定义在区域文件 db. 0-25. 2. 0. 192. in-addr. arpa 内。下面我们将遵循 RFC 2317，但短线也是可以使用的。

这种做法支持将各子网的记录委托给不同的子区域管理员。

让我们看看，在我们的范例情形中，这是如何工作的。在对应于这个2.0.192.in-addr.arpa.区域的父区域文件内，我们将配置如下信息[⊖]。

```
2.0.192.in-addr.arpa. IN SOA dns.ipamworldwide.com.
admin.ipamworldwide.com. ( 1 2h 30m 1w 1d )

$ORIGIN 2.0.192.in-addr.arpa. //implicit(隐式的)
0/25 IN NS dns.A1.ipamworldwide.com. //authoritative servers(权威服务器)
     IN NS dns.A2.ipamworldwide.com. // for 0/25
1    IN CNAME 1.0/25.2.0.192.in-addr.arpa.
2    IN CNAME 2.0/25.2.0.192.in-addr.arpa.
3    IN CNAME 3.0/25.2.0.192.in-addr.arpa.
...
127    IN CNAME 127.0/25.2.0.192.in-addr.arpa.
128/25 IN NS dns.B1.ipamworldwide.com. //authoritative servers(权威服务器)
       IN NS dns.B2.ipamworldwide.com. // for 128/25
129    IN CNAME 129.128/25.2.0.192.in-addr.arpa.
130    IN CNAME 130.128/25.2.0.192.in-addr.arpa.
131    IN CNAME 131.128/25.2.0.192.in-addr.arpa.
...
254    IN CNAME 254.128/25.2.0.192.in-addr.arpa.
```

依据标准的域树遍历过程，当发出查询的名字服务器查询2.0.192.in-addr.arpa.区域的权威DNS服务器时，在相应DNS服务器上如上所见的文件并不提供一个解析，而是下一步骤，将期望的IP地址应答，通过一条CNAME记录，指向另一个FQDN。迄今为止的过程中，对IP地址192.0.2.185的主机名的查询，将得到指向185.128/25.2.0.192.in-addr.arpa的一个CNAME。我们也知道要问谁来解析这条查询，原因是有两条NS记录列为关联域128/25.2.0.192.in-addr.arpa.的权威，即dns.B1.inparworldwide.com和dns.B2.iparmworldwide.com。

在这些服务器上的相应128/25.2.0.192.in-addr.arpa.区域文件将包含如下信息。

```
128/25.2.0.192.in-addr.arpa. IN SOA dns.B1.ipamworldwide.com.
admin.ipamworldwide.com. ( 1 2h 30m 1w 1d )
128/25.2.0.192.in-addr.arpa. IN NS dns.B1.ipamworldwide.com.
128/25.2.0.192.in-addr.arpa. IN NS dns.B2.ipamworldwide.com.
129.128/25.2.0.192.in-addr.arpa. IN PTR public1.ipamworldwide.com.
130.128/25.2.0.192.in-addr.arpa. IN PTR public2.ipamworldwide.com.
131.128/25.2.0.192.in-addr.arpa. IN PTR www.ipamworldwide.com.
```

⊖ 在本书中的DNS配置文件名范例都使用BIND DNS格式[144]

或使用“相对”域名的缩写格式。

@ INSOA dns. B1. ipamworldwide. com. admin. ipamworldwide. com. (1 2h30m 1w 1d)

// Implicit $ORIGIN 128/25. 2. 0. 192. in-addr. arpa.

IN NS dns. B1. ipamworldwide. com.

IN NS dns. B2. ipamworldwide. com.

129 IN PTR public1. ipamworldwide. com.

130 IN PTR public2. ipamworldwide. com.

131 IN PTR www. ipamworldwide. com.

…

185 IN PTR server-x. ipamworldwide. com.

查询这个区域文件，查找这个被索引的 CNAME 别名到 185. 128/25. 2. 0. 192. in-addr. arpa.，我们找到 PTR 记录，它指向关联的主机名 server-x. ipamworldwide. com，这就完成了解析。

对于大于 C 类网络的非字节边界的网络（即/9 ~/15 和/17 ~/23），可使用域别名（DNAME）记录。例如，172. 16. 0. 0/14 网络就可被分配并委托给工程组的一名管理员。有关这个网络的反向查询可指向到工程组的 DNS 服务器，按照如下范例是 dns [1-2] . eng. ipamworldwide. com，被配置在 172. in-addr. arpa. 区域文件内。

16/14. 172. in-addr. arpa. IN NS dns1. eng. ipamworldwide. com

16/14. 172. in-addr. arpa. IN NS dns2. eng. ipamworldwide. com

16. 172. in-addr. arpa. IN DNAME 16. 16/14. 172. in-addr. arpa.

17. 172. in-addr. arpa. IN DNAME 17. 16/14. 172. in-addr. arpa.

18. 172. in-addr. arpa. IN DNAME 18. 16/14. 172. in-addr. arpa.

19. 172. in-addr. arpa. IN DNAME 19. 16/14. 172. in-addr. arpa.

这些表项将所有四个/16 网络（组成工程组的/14 网络）的反向查找委托给他们的 DNS 服务器，由上所示的前两条记录指明。接下来的四条记录将这四个/16 反向域映射到这个被委托的 16/14. 172. in-addr. arpa. 域。

本质上，我们在反向树中插入了一个人为制造的层，用作一个合并（consolidation）点。因此，为了解析 IP 地址为 172. 18. 45. 94 的一台主机的 PTR 记录，解析名字服务器将沿 172. in-addr. arpa. 树向下遍历。向下的下一个节点 18. 172. in-addr. arpa.，依据 DNAME 查找，具有 18. 16/14. 172. in-addr. arpa. 的一个域别名。接下来，通过查询 dns1. eng. ipamworldwide. com 的 DNS 服务器（它是 16/14. 172. in-addr. arpa. 区域的权威服务器），则我们在这个区域内解析相应的 PTR 表项。

94. 45. 18. 172. in-addr. arpa. IN PTR host. eng. ipamworldwide. com.

9. 3. 3 IPv6 反向域

IPv6 反向域映射要有点麻烦。和 IPv4 的情况一样，IPv6 地址必须被反向，依此维持其十六进制格式。但 IPv6 地址首先必须被“填充”到完全的 32 个十六进制数字

形式；即在第 2 章中讨论的两种缩写形式必须被消除，方法是在冒号之间包括前导零，并将双冒号表示的隐含零填充到位。图 9-5 形象地说明了 IPv6 地址 2001：DB8：B7：：A8E1 的一个范例过程。地址必须被扩展或填充，各个数字要做反向处理。之后，必须对这个结果做“域化”处理，方法是消除冒号，在每个数字之间插入点号（.），并附加 ip6. arpa. 顶层域。

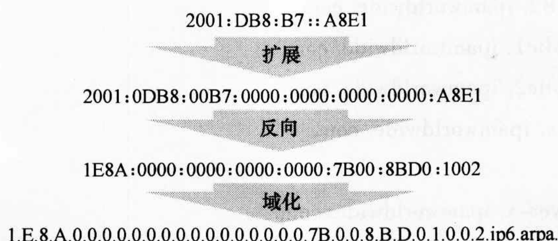


图 9-5 IPv6 地址到反向域的映射

图 9-6 形象地说明了对 IPv6 地址反向的逻辑，目的是将其表示在一个域层次结构中，从左到右读取时，是从比较具体的信息到不太具体的信息顺序的。这可直接类比于图 9-4，那个图形象地说明了用于 IPv4 地址的这个概念。在图 9-6 中所用的完全的 32 个十六进制数字形式，提供了沿 ip6. arpa. 域树向下的一个唯一的（虽然有点长）的遍历（在图中没有画出）。

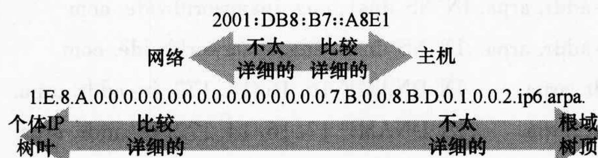


图 9-6 IPv6 反向域表示法

注意这个范例形象地说明了一个完全 128bit IPv6 地址的反向域表示。和 IPv4 中一样，各子网可具有对应的反向域定义。对于一个/64 分配，仅有开始的 64bit（16 个十六进制数字）可被包括在内。因此，对于上面的主机，其/64 子网反向区域表示将被定义为

0. 0. 0. 0. 7. B. 0. 0. 8. B. D. 0. 1. 0. 0. 2. ip6. arpa.

对于在非尼伯边界上分配的 IPv6 网络的反向域表示方法，在 RFC 2317 中没有得到正式解决；但是，在规范中确定的相同技术可被映射到对应于非尼伯边界的 IPv6 地址块分配的 IPv6 反向区域。让我们以范例形象地说明这点。假定北美团队（team）期望从其 2001：db8：4af0：8000：：/52 地址块分配四个/54 地址块，即 2001：db8：4af0：8000：：/54、2001：db8：4af0：8400：：/54、2001：db8：4af0：8800：：/54 和 2001：db8：4af0：8c00：：/54。使用 CNAME 资源记录的方法，将查询器指向负责对应的反向区域的服务器，则 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa 区域文件看起来有点像下面的内容。

8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa. IN SOA dns. ipamworldwide. com.

```
admin. ipamworldwide. com. ( 1 2h 30m 1w 1d )
 $ ORIGIN 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa. //implicit(隐式的)
0/54 IN NS dns. A1. ipamworldwide. com. //authoritative servers(权威服务器)
     IN NS dns. A2. ipamworldwide. com. // 对于 2001:db8:4af0:8000::
     /54
0    IN CNAME 0. 0/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
1    IN CNAME 1. 0/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
2    IN CNAME 2. 0/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
3    IN CNAME 3. 0/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
4/54 IN NS dns. B1. ipamworldwide. com. //authoritative servers(权威服务器)
     IN NS dns. B2. ipamworldwide. com. // 对于 2001:db8:4af0:
8400::/54
4    IN CNAME 4. 4/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
5    IN CNAME 5. 4/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
6    IN CNAME 6. 4/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
7    IN CNAME 7. 4/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
8/54 IN NS dns. C1. ipamworldwide. com. //authoritative servers(权威服务器)
     INNSdns. C2. ipamworldwide. com. //对于 2001:db8:4af0:8800::/54
8    IN CNAME 8. 8/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
9    IN CNAME 9. 8/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
a    IN CNAME a. 8/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
b    IN CNAME b. 8/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
c/54 IN NS dns. D1. ipamworldwide. com. //authoritative servers(权威服务器)
     INNSdns. D2. ipamworldwide. com. //对于 2001:db8:4af0:8c00::/54
c    IN CNAME c. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
d    IN CNAME d. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
e    IN CNAME e. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
f    IN CNAME f. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.
```

遵循标准域树遍历过程，当发出查询的名字服务器查询 8. 0. f. a. 4. 8. b. d. 0.1. 0. 0. 2. ip6. arpa. 区域的权威 DNS 服务器时，在对应 DNS 服务器上的上述文件，提供的不是一个解析，但下一步，通过一个 CNAME 记录，将期望的 IPv6 地址答案指向另一个 FQDN。在迄今为止的过程中，请求 IP 地址为 2001: db8: 4af0: 8d03:: f6 的主机名的一条 PTR 查询，得到指向 d. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa 的一个 CNAME。我们也知道为解析这条查询要问谁，原因是有两条 NS 记录被列为这个域的权威，即 dns. D1. ipamworldwide. com 和 dns. D2. ipamworldwide. com。

在这些服务器上对应的 d. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa. 区域文件，将包含如下内容。

```
c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa. IN SOA dns. D1. ipamworldwide. com.
```

admin. ipamworldwide. com. (1 2h 30m 1w 1d)

IN NS dns. D1. ipamworldwide. com.

IN NS dns. D2. ipamworldwide. com.

1. 0. b. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. c IN PTR public1. ipamworldwide. com.

0. 2. 0. a. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. c IN PTR public2. ipamworldwide. com.

f. c. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. d IN PTR www. ipamworldwide. com.

…

6. f. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. d IN PTR server-y. ipamworldwide. com.

为这个索引的 CNAME 别名，即 6. f. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. d. c/54. 8. 0. f. a. 4. 8. b. d. 0. 1. 0. 0. 2. ip6. arpa.，查询这个区域文件，我们发现 PTR 记录指向关联的主机名 server-y. ipamworldwide. com，这就完成了解析。

9.3.4 其他区域

（1）根线索。在解析过程的综述中，我们提到了一个线索文件。这个文件应该提供 DNS 服务器名和地址的一个列表（以 NS、A 和 AAAA 资源记录的形式提供），如果解析器查询不能通过权威的、转发的或缓存的数据得到解析，则服务器应该查询这个列表。典型情况下，线索文件将列出因特网根服务器（多台服务器），它们是域树之根（.）的权威服务器。为了定位要解析查询的一台权威服务器，对一台根服务器的查询的做法，使发出查询的服务器从顶部开始，沿域树向下开始遍历。因特网根服务器的根文件内容可从 www. internic. net /zones/ named. root 得到，虽然 BIND 和微软 DNS 服务器实现在它们的发布时就包括了这个文件。

如我们将在第 11 章讨论的，一些环境会要求使用根服务器的一个内部集合，其中因特网访问受到组织机构策略的限制。在这种情形中，可使用线索文件的一个内部版本，它列出内部服务器（而不是因特网根服务器）的名字和地址。组织机构自己将需要维护内部根服务器的列表及其必要的（requisite）根区域配置。

（2）本地主机区域。证明必不可少的另一个区域文件是本地主机区域。本地主机区域使在给定服务器上将“localhost”解析为一个主机名。一个相应的 in-addr. arpa. 区域文件解析 127. 0. 0. 1 回环地址。在 0. 0. 127. in-addr. arpa 区域内的单一表项将地址 1 映射到主机自己。需要这个区域的原因是，对于 127. in-addr. arpa 域或子域，是不存在上游（upstream）权威的。类似地，对于相应的 IPv6 回环地址：: 1，需要定义 IPv6 等价物。主机名区域简单地将本地主机名映射到其 127. 0. 0. 1 或：: 1 等 IP 地址，分别使用一条 A 和一条 AAAA 记录。

9.4 解析器配置

就像 DHCP 事务一样，DNS 解析发生在后台，并涉及一个客户端和一个服务器。理想情况下，终端用户甚至都不知道它发生过；他们键入一个 web 地址并连接。解析器软件必须配置有解析时要查询的 DNS 服务器（可能是多台服务器）。DHCP 不要求

初始客户端配置（原因是它只需简单地向一个周知的地址广播或组播即可），因此，与此不同的是，在使用之前，DNS 确实需要某种基本的客户端配置。这种初始配置可采取人工完成或从一台 DHCP 服务器得到这个信息来完成。

图 9-7 形象地说明了一个微软 Windows 解析器的配置，它采取人工定义要查询的 DNS 服务器，或使用 DHCP 自动地得到 DNS 服务器地址。

微软 Windows 在其图形界面内可配置要查询的多台 DNS 服务器表项。注意，在图 9-7 右侧的屏幕上所示的“蛮力”（brute force）方法中有两个表项，一个表项用于首选，另一个表项是替代表项。以特定顺序单击高级标签（tab）就激活了两个以上的表项。我们建议，为解析器至少要配置两台 DNS 服务器，以便预防一台 DNS 服务器不能工作的情形，这时解析器将自动地查询一台替代服务器。如果“自动得到 DNS 服务器地址”（Obtain DNS server address automatically）单选按钮被选中，如图 9-7 所示，则解析器将通过 DHCP 得到 DNS 服务器的一个列表。

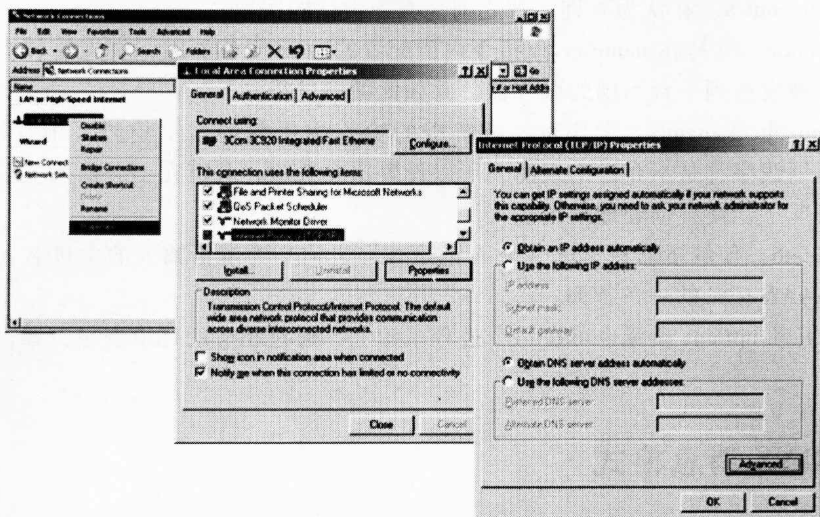


图 9-7 Microsoft Windows 操作系统上，要被查询的 DNS 服务器 IP 地址配置

在基于 UNIX 或 Linux 操作系统上，/etc/resolv. conf 文件是可编辑的，以此配置解析器。在这个文件中的关键参数是指向 DNS 服务器的一个或多个 nameserver 声明语句，但许多选项和附加指令（directive）支持进一步的配置细化，如下描述。斜体文本应该由实际索引的数据替换，例如，*domain* 应该以一个 DNS 域名替换。

（1）nameserver *IP_address*。查询名字解析的一台递归 DNS 服务器的 IP 地址；允许并鼓励有多个 nameserver 表项。nameserver 表项指令解析器到哪里去指向（direct）DNS 查询。

（2）domain *domain*。DNS 域，是这台主机所在的域，在主机上安装了这个解析器。相对于完全合格的主机域名而言，当解析相对主机名时使用这个选项。

（3）search *domain*（*s*）。多达六个域的搜索列表，到这些域搜索输入的主机名，以便进行解析。因此，如果我们键入 www 进行解析，则解析器将后续地将这

个参数中配置的域附加其后，以便尝试解析该查询。如果在 resolv. conf 中存在表项 search ipamworldwide. com.，表项 www 将得到 www. ipamworldwide. com 的一个解析尝试。

（4） sortlist *address/mask list*。依据地址/掩码组合的指派列表，支持被解析 IP 地址的排序。这使在针对一条查询返回多个 IP 地址时，解析器可选择一个“较近的”目的地。

（5） options。在下面各项参数之前的关键字，这些参数支持规范相应的解析器参数，包括如下参数。

1） debug。打开调试。

2） ndots *n*。解析器将考虑分析之前，在所要求输入名内部，为点号数定义一个阈值，被输入的名字简单地是一个主机名或一个合格的域名。当考虑一个主机名时，将被查询的主机名，其后将附加 domain 或 search 参数内确定的域名。

3） timeout *n*。在认为查询失败之前，查询尝试的次数。

4） rotate。支持在 nameserver 指令内所配置的 DNS 服务器间，实施轮转查询。每次查询将被发送到一台不同的服务器，并如此循环进行。

5） no-check-names。关闭对要被解析的输入主机名的名字检查。正常情况下，例如，下划线字符是不允许出现的，所以设置这个选项，就可在不对输入主机名进行验证的情况下，使查询处理继续进行。

6） inet6。使解析器在尝试一个 A 记录查询之前，为解析输入的主机名，发出针对一个 AAAA 记录的一条查询。

search 和 options 设置也可在每个进程基础上，通过相应的环境变量设置，并加以覆盖。

9.5 DNS 消息格式

9.5.1 域名的编码

到此为止，我们讨论了如下内容：将 DNS 信息组织成一个域层次结构，一个客户端或解析器如何实施解析的基础知识。后者的做法是向一台 DNS 服务器发出一条递归查询，DNS 服务器接下来依据域层次结构，迭代地发出查询，目的是得到查询的答案。接下来我们将较深入地挖掘 DNS 查询消息格式和通用的消息格式，但首先我们将介绍 DNS 消息内部域名的表示。域名被格式化为标签的一个序列。各标签由如下各项组成：一个字节的长度字段，后跟表示标签本身的该数量（长度）的字节/美国信息交换标准码（ASCII）字符。这个标签序列以长度字段为 0 表示根“.”域的方式终结。例如，www. ipamworldwide. com. 的标签序列看起来将像如下的 ASCII 格式，其中长度字节以图 9-8 中较黑阴影突出显示。

以左上开始，第一个长度字节的数值“3”，指明后跟的三个字节组成第一个标签“www”。在这个标签之后的第五个或下一个字节是我们的下一个长度字节，它具

有数值“13”（0xD），它是“ipamworldwide.”的长度。在这个标签之后，后跟字节的数值“3”是“com.”的长度。最后，为零值的字节指明根“.”域，这就完整地使域名符合要求了。注意，图中的较黑阴影字节被编码为长度字节，以便将主机或包含数字的域名字符区分开来。一个名字中的第一个字节将几乎总是[⊖]一个长度字节，后跟那么多字节（长度表示的），来表示第一个标签的，并为了去除二义性而直接后跟另一个长度字节。

0 bit 7	8 15	16 23	24 31
3	W	W	W
13	I	P	A
M	W	O	R
L	D	W	I
D	E	3	C
O	M	0	

图9-8 DNS标签

9.5.2 名字压缩

一个给定的DNS消息可包含多个域名，其中许多域名可能具有重复的信息，例如ipamworldwide.com.后缀。DNS规范支持消息压缩，目的是降低重复信息，因此就降低了DNS消息的尺寸。这是如下方法发挥作用的，方法是使用到DNS消息内部其他位置的指针，该DNS消息确定了一个通用的域后缀。之后将这个域后缀附加在由指针索引位置的位置点。

让我们举一个例子，即我们对www.ipamworldwide.com.的查询返回一对DNS服务器，可用其查询更多的信息：ns1.ipamworldwide.com.和ns2.isp.com.。两个答案之一（第一个答案和第二个答案），这些域名的ipamworldwide.com.部分对于查询是相同的，而对于其他查询仅有.com部分是相同的。因此，消息是如下形成的，方法是完整地确定域名www.ipamworldwide.com.，如图9-8所示。之后，当确定ns1时，并不完整地确定ns1.ipamworldwide.com，仅确定ns1，后跟指向消息中前面ipamworldwide.com.后缀的一个指针。当识别ns2.isp.com时，确定ns2.isp标签，后跟指向消息内部.com后缀的一个指针。

DNS解析器和服务器如何将一个指针与一个标准的标签长度字节做出区分呢？DNS标准规定每个标签的长度为0~63个字节。以二进制表示，就是00000000~00111111。因此，前2bit，在这种情形中是$[00]_2$，将该字节标识为一个标准长度字节，指明后跟标签的长度。通过将前2bit设置为$[11]_2$，就可识别一个指针，它由两个字节组成，其中$[11]_2$后跟14bit，识别从DNS首部开始的字节偏移。DNS消息首部的第一个字节被看做字节0，当产生消息时，指针是这样定义的，即从这个点开始的字节偏移。

⊖ 正如我们接下来将讨论的，长度字节也可由一个两字节指针或一个DNS扩展标签组成。

图 9-9　采用指针的名字压缩

让我们看看这是如何从我们前边的例子中进行映射的。让我们假定从 DNS 首部的第 12 字节开始，我们包括域名 www. ipamworldwide. com。现在，在消息的后面，从首部开始的第 56 字节起，我们希望对响应 ns1. ipamworldwide. com 和 ns2. isp. com 进行编码。

图 9-9 说明了这看起来是什么样的。如我们在前面讨论的，第一部分是长度字节（黑色阴影）后跟对应数量的字节（亮色阴影）。在我们的例子中，在字节位置 56 处，名字的 ns1 部分被正常地编码，使用的是标签长度“3”，后跟 ns1。但是，下一个字节不是一个标准的长度字节，而是一个“双字节”的指针，原因是它以 $[11]_2$ 开始，并显示为图中的黑色阴影。在该指针的 14bit 偏移字段中，编码的数值是“16”，指明域名的那部分开始于从 DNS 首部开始的第 16bit 字节偏移处，这部分域名应该附加在已经确定的 ns1 标签之后。下图中的第一行字节枚举了各字节偏移（斜体），第 16 字节是数值为“13”的长度字节，后跟 ipamworldwide 的编码，再后跟数值为“3”的一个长度字节，之后是 com，再后是“.”（数值“0”的长度字节）。将这些串接在一起，我们得到结果：ns1. ipamworldwide. com。

在处理该指针之后，返回到下一个域名，我们寻找 ns2. isp 的编码，后跟指向字节偏移 30 的⊖一个指针，它指向字节长度“3”，后跟 com.，这就完成了域名 ns2. isp. com。仅考虑这三个范例域名，在消息中由域名占据的字节数可被压缩，范围是 59 ~ 39 个字节。

9.5.3　国际域名

DNS 解析器和服务器以 ASCII 格式的消息传递主机查询和响应。配置信息是以 ASCII⊖文本文件形式存储的。不幸的是，ASCII 字符已经定义用来有效地表示英语，它们不能支持其他语言字符的格式化，特别是使用非拉丁字母的那些语言。这个限制当然影响了不使用英语的国家的国民方便使用 IP 应用的便利性。RFC 3490[94] 是一个

⊖ 注意上面以黑色所示的双字节指针，被显示为其字节方式的十进制数表示，在我们的例子中，按照惯例显示为第 2 字节中以十进制表示的偏移。但仅明显地陈述一下，当解析一个指针时，不要仅依赖于这第 2 字节，原因是一个指针数值可从 0 到 $2^{14}=16384$，在这个最大值处，十进制表示将为 255-255。

⊖ RFC 2673[182] 最初是一个标准跟踪方式的 RFC，它定义了在 DNS 名字内使用二进制数据，但 RFC 3363[183] 将 RFC 2673 重返置为（reverted）试验状态。

标准跟踪 RFC，它解决了这项限制㊀。

该 RFC 被称作应用中的域名的国际化（IDNA）。名字中“应用中”这个修饰语暗示了在这个过程中涉及应用。确实，对应用（例如浏览器或电子邮件客户端）施加了责任（onus），它将用户的母语项转换为一个基于 ASCII 的字符串，为进行解析而将其传递到一台 DNS 服务器上。这种巧妙的方法使应用层针对终端用户而支持国际字符集成为可能，而并不影响 DNS 协议（或其他基于 ASCII 的 IP，如 SMTP）。现有 DNS 服务器可被配置成解析这些 ASCII 编码的域名，就像解析原本基于 ASCII 的域名一样。

国际字符集被编码为统一编码（Unicode）字符。依据 Unicode（单一码）联盟（Consortium）网站（www. unicode. org），Unicode 标准“为每个字符提供了一个唯一数，而不管是什么平台、什么程序以及什么语言”。每个字符被表示为一个唯一的 2 或 3 字节十六进制数。RFC 3490 及其相关的 RFC 3491[95]、3454[96] 和 3492[97]，描述了将一个基于 Unicode 的域名转换为一个 ASCII 格式域名的过程。注意从技术角度而言，域标签是每个分别转换的，而并不是整个“域名”转换的。

为了解析国际域名，一台 DNS 服务器必须被配置带有以 ASCII 格式编码的资源记录，特别是 Unicode 映射的 ASCII 字符，它被称作弱码（punycode）。弱码算法的输出得到一个 ASCII 字符串，之后以 ASCII 兼容编码（ACE）首部 xn--作为前缀。因此，在 DNS 基础设施内，表示为 xn-- <附加 ASCII 字符>的域可能是一个国际域名的弱码表示。应用（例如网页浏览器）负责将用户输入的 URL 转换为 Unicode 格式，之后再转换弱码。弱码域名被传递到客户端上的解析器，通过 DNS 使用 ASCII 字符进行解析。在 RFC 3492 中规范了弱码算法，几个 web 站点可用于将表项转换到 DNS。

考虑一个范例[98]：让我们考虑在 żdźblo. com 域中作为 www. żdźblo. com 的一台 web 服务器主机的地址。该域名包含变音符号，并具有 ASCII 字符集外的字符。输入这个 URL 的网页浏览器将此域名转换为 ASCII 字符或弱码为 www. xn--dbo--iwalzb. com。在 DNS 中 www. xn--dbo-iwalzb. com. 主机的对应 A 或 AAAA 记录表项将使终端用户能够输入一个母语的 URL，同时利用部署在世界各地的现有 DNS 服务器基础，通过目的地 web 服务器的 IP 地址来识别和连接该服务器。净结果是，在通信导线上（on the wire）发送的这些 DNS 消息被编码为 ASCII 字符。

9.5.4　DNS 消息格式

现在让我们更详细地看看用来实施这个整体解析功能的 DNS 消息格式，它将我们较早讨论的标签格式的域名集成在了一起。DNS 消息默认地在 UDP 上传输的，使用端口 53。TCP 也可在端口 53 上使用。一条 DNS 消息的基本格式如图 9-10 所示。

（1）消息首部包含各种字段，这些字段定义了消息的类型和关联的信息，其中

㊀ 注释：定义“IDNA2003”的 RFC 3490 已被 RFC 5890-4[184-188] 这四个 RFC 所更新，被称作“IDNA2008”，这每个版本均有规范工作开始的年份指示。在这些版本间存在一些差异，但一般而言，本节中的资料对它们均适用。

包括如下每个字段的记录数。

（2）Question（问题）节确定通过这条消息被查找的信息。

（3）Answer（答案）节包含零个或多个资源记录，它们回答问题节中确定的查询。

（4）Authority（权威）节包含零个或多个资源记录，索引给定答案的权威名字服务器，或指向沿树向下的委托名字服务器，后续的迭代查询可向此服务器发出。

（5）Additional（附加）节包含零个或多个资源记录，包含与问题相关的附加信息，但未必是问题的严格答案。

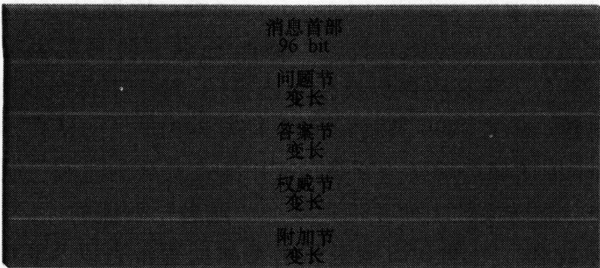


图 9-10 DNS 消息字段[99]

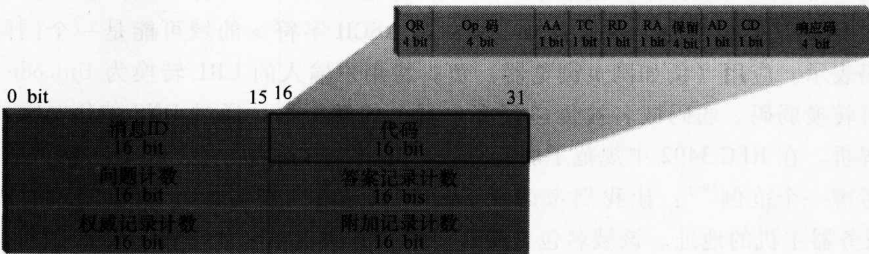


图 9-11 DNS 消息首部[99]

1. 消息首部

在每个 DNS 消息上包括 DNS 消息首部，并传递要包含的消息类型和相关联的参数，如图 9-11 所示。

消息首部由六个 16bit 字段组成。

（1）消息 ID。也称作事务 ID，这是解析器指派的一个标识符，并从 DNS 服务器复制到应答中，这使解析器将响应与查询关联起来。

（2）代码。与这条消息关系密切的消息代码。我们接下来详细研究这些代码字段。

（3）问题计数（QDCOUNT）。在 DNS 消息的问题节中包含的问题数。

（4）答案记录计数（ANCOUNT）。在 DNS 消息的答案节中包含的资源记录数。

（5）权威记录计数（NSCOUNT）。在 DNS 消息的权威节中包含的资源记录数。

（6）附加记录计数（ARCOUNT）。在 DNS 消息的附加节中包含的资源记录数。

定义了如下代码比特。

（1）QR（查询/响应）。这个标志表明这条消息是一条查询（0）还是一条响应（1）。

（2）Opcode（操作码）。这条消息的操作码。目前，定义了如下数值：

1）0 = 查询。

2）1 = 保留（以前是反向查询，目前废弃不用）。

3）2 = 服务器状态请求。

4）3 = 保留。

5）4 = 通知——使一台主区域服务器通知拥有同一区域的一台从属区域服务器（从属服务器要确认），对区域数据作出了改变。对于通知消息，权威节和附加节是不用的，在 DNS 首部中响应的记录计数应该设置为 0。

6）5 = 更新——使一个客户端或 DHCP 服务器更新一台 DNS 服务器上的区域数据。对于更新消息，DNS 消息字段和对应首部字段的解释和上述的有所不同。在下一节描述更新消息的消息格式。

7）6 ~ 15 = 未指派

（3）AA（权威答案）。当设置时，这条消息包含了问题的一个权威答案。这意味着响应是从一台 DNS 服务器得到的，该服务器配置有区域的信息。如果没有设置的话，则答案是从一台非权威 DNS 服务器得到的，极可能是以前查询的被缓存信息。当提供多个答案时，这个标志与答案节中的第一条记录有关。当在查询上由客户端设置时，这表明要求得到一个权威答案（不被缓存的）。

（4）TC（截短的响应）。这个码表明这条消息由于传输的原因而被截短。一般而言，这是由于 UDP 报文的报文长度限制导致的，UDP 是 DNS 使用的默认传输层协议。

（5）RD（期望采用递归法）。这个标志指明，查询者将希望 DNS 服务器迭代地解析查询，必要时遍历域树。多数解析器设置这个标志，指明一个查询为一个递归查询，同时一般来说，当查询其他服务器时，一台 DNS 服务器不会设置这个标志。

（6）RA（递归法是可用的）。这个标志指明这台 DNS 服务器可支持递归查询法。

（7）保留或 *Z*bit。保留的（0）。

（8）AD（可信的数据）。在 DNS 安全扩展（DNSSEC）上下文内使用，由一台名字服务器设置这个比特，用来指明在答案节和权威节内的信息是可信的，这意味着它已经过认证。

（9）CD（检查被禁止）。用于 DNSSEC 上下文内，在一台 DNSSEC 名字服务器处理这个特定的查询过程中，这个比特使一个 DNSSEC 解析器能够禁止签名验证。

（10）响应码（RCODE）。向客户端提供结果状态。表 9-1 中汇总了当前定义的响应码。注意，给定 4bit 的 RCODE 字段，则十进制数值 1 ~ 15 就可编码在 DNS 首部 RCODE 字段内。

DNS 扩展（EDNS0，在本章后面讨论）OPT（OPTion，选项）资源记录为容量增加了 8 个附加的 RCODE 比特，当与首部 RCODE 比特一起使用时，这种做法将总数增加为 12bit（达十进制数值 4095）。

您将注意到十进制 16 有两种解释。当编码在 OPT 资源记录内时，解释为 BADVERS，而当编码在一个 TKEY 或 TSIG 资源记录内时，结果解释为 BADSIG。

表 9-1 DNS 消息响应码①[106]

RCODE		名字	描述	文献
十进制	十六进制			
0	0	NoError	没有错误	RFC 1035[99]
1	1	FormErr	格式错误——服务器不能解释查询	RFC 1035[99]
2	2	ServFail	服务器故障——服务器问题导致这条查询不能被处理	RFC 1035[99]
3	3	NXDomain	不存在的域——域名不存在	RFC 1035[99]
4	4	NotImp	没有实现——这台服务器不支持该查询类型	RFC 1035[99]
5	5	Refused(拒绝的)	查询被拒绝——服务器拒绝所请求的查询,例如拒绝一个区域传递请求	RFC 1035[99]
6	6	YXDomain	当在 DNS 更新前提条件处理过程中确定时,不应该存在的名字却是存在的	RFC 2136[100]
7	7	YXRRSet	当在 DNS 更新前提条件处理过程中确定时,不应该存在的 RRSet 却是存在的	RFC 2136[100]
8	8	NXRRSet	当在 DNS 更新前提条件处理过程中确定时,应该存在的 RRSet 却是不存在的	RFC 2136[100]
9	9	NotAuth	服务器不是 DNS 更新消息的区域节中所列区域的权威服务器	RFC 2136[100]
10	A	NotZone(不是区域)	在前提条件或一条 DNS 更新消息更新节中所用的名字,没有被包含在该消息区域节所指明的区域之中	RFC 2136[100]
11 ~ 15	B ~ F	可用于指派	—	—
16	10	BADVERS	不被支持的(不良)OPT RR 版本	RFC 2671[101]
16	10	BADSIG	TSIG 签名失效	RFC 2845[102]
17	11	BADKEY	不可识别的密钥(key)	RFC 2845[102]
18	12	BADTIME	签名超出了有效的服务器签名时间窗口	RFC 2845[102]

（续）

RCODE		名字	描述	文献
十进制	十六进制			
19	13	BADMODE	无效的 TKEY 模式——这台服务器不支持所请求的模式	RFC 2930[103]
20	14	BADNAME	不存在的或重复的密钥名	RFC 2930[103]
21	15	BADALG	不支持的算法	RFC 2930[103]
22	16	BADTRUNC	不良的截短操作——消息认证码（MAC）太短	RFC 4635[104]
23 ~ 3840	14 ~ F00	可用于指派	—	
3841 ~ 4095	F01 ~ FFF	为专用用途保留	—	RFC 5395[105]

① 如果您查阅 IANA 网站（www. iana. org/assignments/dns-parameters），则您将注意到 4095 以上的数值。从技术角度而言，这些不是 RCODE，仅反映 TSIG 和 TKEY 元资源记录类型内的 16bit 错误字段，它为这两个资源记录类型提供高达 65535 的容量。

2. 问题节

如您可能猜到的，DNS 消息格式内的问题节，包含这条查询要询问的问题。该节可包含一个以上的问题，由 QDCOUNT 首部字段中所指数字标明。每个这样的问题都有如下格式（见图 9-12）。

QNAME 字段包含域名，格式化为一系列标签。QType 字段指明查询类型，或问这个问题的目的是什么。可包括任意资源记录类型，我们将在下一章详细讲述。但是，对于请求区域传递，存在一些独特的 QType 数值，例如当前定义的那些，包括如下内容（见表 9-2）。

QCLASS 字段指明这个查询是针对哪一类的，例如因特网类的 IN，它是最常见的类。本质上而言，类使平行名字空间的管理成为可能。一般而言，当前定义的 QCLASS（和 DNS CLASS）定义在表 9-3 中。

3. 答案节

答案节以资源记录的形式，包含零个或多个答案。答案的数量在 ANCOUNT 首部字段中确定。在下一章中，我们将讨论不同类型的资源记录，它们都使用一个通用的格式，如图 9-13 所定义的。

名字字段，也称作属主名字字段，是对应于这个资源记录的查询名（对应于原始问题中的查询值或 QNAME）。

0 bit 15 16 31

QNAME
可变比特数

QTYPE
16 bit

QCLASS
16 bit

图 9-12 问题节格式[99]

表 9-2　DNS QType①[106]

仅是 QType	查询目的	QType ID(十进制)	IETF 状态	定义该类型的文档
*	所有资源记录	255	标准	RFC 1035
MAILA	邮件代理资源记录	254	试验型的	RFC 1035
MAILB	邮箱资源记录	253	过时的	RFC 1035
AXFR	绝对区域传递(整个区域)	252	标准	RFC 1035
IXFR	增量区域传递(仅涉及发生的变化)	251	建议的标准	RFC 1995

① 另外，表 12-1 中的 RRType 可用作 QType。

类型字段指明可为这个名字提供的信息类型。例如，类型 A 意味着这个资源记录为给定名字提供 IPv4 地址信息。在下一章中讲解资源记录类型，并汇总于表 10-1。

类（Class）字段表示名字空间类，例如用于因特网的 IN。表 9-3 给出了有效的类。

TTL 或存活时间字段以秒为单位，给出了资源记录有效寿命的一个时间数值。这个消息的接收者可将此信息缓存 TTL 秒，并在此时间内可靠地（不用担忧地）使用这个信息。但是，在 TTL 超期时，应该丢弃被缓存的信息，并发出一条新的查询。

表 9-3　DNS 类[106]

类(CLASS)		名字	描述	参考文献
十进制	十六进制			
0	0	保留	保留	RFC 5395
1	1	IN	因特网	RFC 1035
2	2	未指派	未使用	IANA
3	3	CH	Chaos(混乱)	RFC 1035
4	4	HS	Hesiod	RFC 1035
5 ~ 253	5 ~ FD	未指派	未使用	IANA
254	FE	NONE	无	RFC 2136
255	FF	*(任意)	任一类(作为 QCLASS 是有效的,但不在资源记录上)	RFC 1035
256 ~ 65 270	100 ~ FEFF	未指派	未使用	IANA
65 280 ~ 65 534	FF00 ~ FFFE	为专用用途保留	—	RFC 5395
65 535	FFFF	保留	保留	RFC 5395

RDLENGTH 字段指明了结果（RDATA）字段的长度，是以字节为单位的。对于给定属主，RDATA 字段包含了所识别类中所确定类型的相应信息。如我们将看到的，当详细研究多样化的资源记录类型时，RDATA 字段有一个变形的格式。

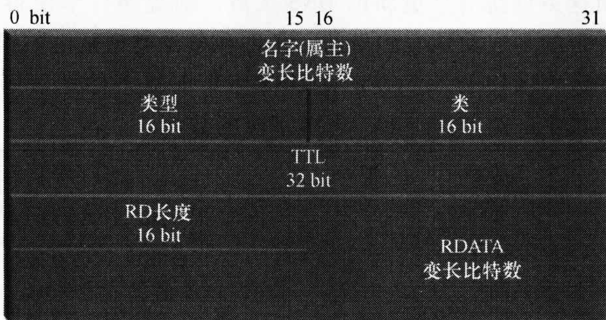


图 9-13 答案节格式[99]

4. 权威节

权威节包含了 NSCOUNT 数量的答案，和在答案节中讨论的一样，它使用相同格式的资源记录。一般来说，在权威节内，仅有 NS（名字服务器）资源记录是有效的，虽然当被查询的名字服务器是权威的，但答案节为空时，多数名字服务器在该节中返回一个 SOA（Start of Authority，起始授权机构）记录。本节也包含有关其他名字服务器的信息，这些服务器是被查询信息的权威服务器。这个信息由查询解析器使用，或更可能被递归名字服务器所用，以便在寻找最终答案而遍历域树时，确定要查询的下一台名字服务器。

5. 附加节

附加节以资源记录的形式，包含 ASCOUNT 数量的答案，它提供查询的附加的或有关的信息，与答案节中讨论的格式相同。

9.5.5 DNS 更新消息

更新消息使一台客户端、DHCP 服务器或其他源能够在一个区域内实施一个或多个资源记录的一条更新（增加、修改或删除）。虽然更新消息利用刚描述过的 DNS 消息的相同基本格式，但一些字段的解释是不同的。更新消息，在 DNS 消息首部中以操作码 =5 表示，其编码如下（见图 9-14）。

将这个格式与图 9-10 所示的非更新 DNS 消息比较。消息首部与“正常”DNS 消息的格式相同，但其他各节的解释不同。



图 9-14 DNS 更新消息格式[100]

区域节识别由这条更新消息更新的 DNS 区域。前提条件节使得必须被满足的条件确定成为可能，目的是成功地实施更新。条件和条件类型是由前提条件节内每个资源记录编码参数的数值确定的。下表定义 DNS 更新前提条件是如何解释的，它的依据是前提条件节内属主、类、类型和 RData 字段的数值。

属主	类	类型	RData	前提条件解释
匹配	ANY[255]	ANY	空	要匹配的属主名字在这个区域中已被使用
匹配	ANY[255]	匹配	空	具有匹配属主和类型的一条 RRSet 是存在的(数值无关的，即任意 RData 都匹配)
匹配	NONE[254]	ANY	空	要匹配的属主名字在这个区域中没被使用
匹配	NONE[254]	匹配	空	带有要匹配属主和类型的一条 RRSet 在这个区域中不存在
匹配	与区域类相同	匹配	匹配	带有要匹配属主、类型和 RData 的一条 RRSet 存在于这个区域中(数值相关的，即 RData 匹配)

更新节包含要添加到区域或从区域删除的资源记录，使用的是如下前提条件节所用的一种类似编码。

属主	类	类型	RData	更新的解释
属主添加	与区域类相同	RR 类型	RR RData	将确定的属主、类型和 RData 的这条资源记录(可能是多条记录)添加到区域的 RRSet
属主删除	ANY[255]	RR 类型	空	将确定属主和类型的资源记录删除
属主删除	ANY[255]	ANY	空	将确定属主名的所有资源记录删除
属主删除	NONE[254]	RR 类型	RR RData	从区域中删除确定属主、类型和 RData 的资源记录(可能有多条记录)

附加数据节包含与这条更新有关的资源记录，例如区域外黏结（out of zone glue）记录。

考虑带有前提条件和以如下编码的更新字段，接收一条更新消息的例子如下。

字段	属主	类	类型	RData
前提条件	host. ipamworldwide. com	IN	DHCID	H8349a +) 3jELeA = = ES1
更新	host. ipamworldwide. com	IN	A	10. 0. 0. 200

更新节的内容将仅当满足前提条件时才被考虑。在这种情形中，前提条件是 host. ipamworldwide. com. IN DHCID H8349a +) 3jELeA = = ES1 记录存在于区域中，即前提条件类型 RRSet 带有匹配属主、类型和 RData（数值依赖）。如果确实存在，那么来自更新节的 host. ipamworldwide. com. IN A 10. 0. 0. 200 资源记录将被添加到区域之中。如果不存在，则不执行更新。

这个特定的例子形象地说明了 ISC DHCP 服务器，在指派一个 IP 地址（在此情形中是向 host. ipamworldwide. com. 指派 10. 0. 0. 200）时，如何实施 DNS 数据的动态更新。DHCID 记录提供了接收 IP 地址的主机硬件地址的一个散列值，以此唯一地识别

该主机。更新该地址记录的前提条件，提供了确保仅有这条 A 记录的原始持有者才能修改它的一种方法，这就使命名重复或劫持风险最小化。

9.5.6 DNS 扩展（EDNS0）

迄今为止，在我们讨论 DNS 消息首部时，人们会观察到所有代码比特都被指派，但仅有一个代码比特，且是附加的响应代码指派在必要时才需要。另外，相比于原始确定的尺寸限制 512B，许多主机可处理较大型的多部分（multi-part）组成的 UDP 报文。作为这些限制的一个结果，以及期望添加附加的域名标签类型，在 RFC 2671[101] 中定义了 DNS 扩展。

RFC 2671 定义了 DNS 扩展机制的版本 0，表示为 EDNS0。通过定义如下扩展，该 RFC 解决了上述约束。

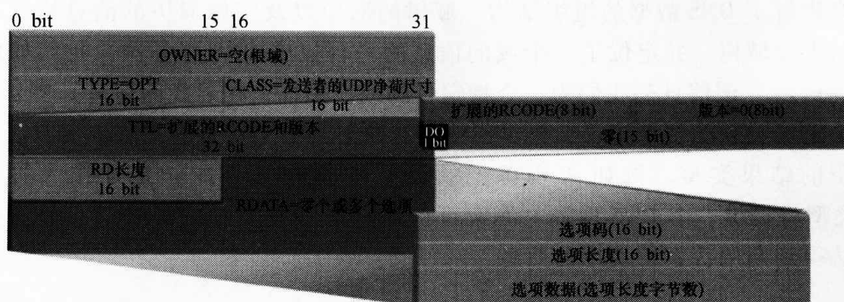


图 9-15 EDNS0 格式[101]

1）为表示 DNS 扩展，定义了一个新的域标签类型。正如我们讨论过的，域标签的前 2bit 唯一地将该标签识别为一个长度字节（前 2bit = $[00]_2$）或一个指针（前 2bit = $[11]_2$）。扩展标签类型被指派 $[01]_2$ 作为它的前 2bit。

2）EDNS0 定义了一个伪资源记录，即 OPT 记录（即 RRType = OPT）。解析器或服务器将 OPT 记录放置在附加节中，目的是通告其相应的能力。OPT 资源记录被用来将发送者（客户端或服务器）的能力通告给接收者，且仅应该存在一个 OPT 记录。

OPT 伪资源记录按如下编码（见图 9-15），这使确定发送者的 UDP 报文尺寸和附加的响应代码比特成为可能。

OPT 记录永远不应出现在一个区域文件中。因此，和其他资源记录一样，当 OPT 伪资源记录利用相同的传输（wire）格式时，则标准字段的定义就已作修改，以便仅提供扩展信息。对于 OPT 记录，NAME 字段为零。TYPE 为 OPT，CLASS 字段指明发送者 UDP 净荷的最大尺寸。32bit 的 TTL 字段被分成如下三个字段。

1）扩展的响应码。将 8bit 添加到 DNS 消息首部的 4bit RCODE，提供了总共 12bit 的字段。

2）EDNS 版本号。

3）扩展的首部标志。bit0 当前被定义为“DNSSEC 答案正常”，这意味着查询服务器能够处理 DNSSEC 资源记录。扩展首部的其他 15bit 当前是保留的。

RDLength 字段指明 RData 字段的长度，它由零个或多个选项组成，每个选项编

码为一个选项码、选项长度和选项值。

迄今为止，一个选项是通过 RFC 5001[191] 进行官方定义的：名字服务器识别符（NSID）选项。由选项码 = 3 定义的这个选项，使一个解析器可发出请求，使一台服务器提供它的身份，依据服务器管理员定义的，即将其名字、IP 地址、伪随机数或其他字符串（在 BIND 中是使用 server-id 语句，可进行配置的）定义为其身份。对于在如下环境中排错，这个 EDNS0 选项是有用的，其中许多台服务器共享同一个 IP 地址，例如在部署任意播寻址的情况或采用负载均衡器的情况。另外两个选项，长时间存活的查询㊀（LLQ；选项码 = 1）和更新租赁寿命㊁（UL；选项码 = 2）目前作为 RFC，还处于停用状态，就这些设置，还没有官方信息发布。

9.5.7 资源记录

本章讲解了 DNS 数据的组织结构、域树的遍历以及实施遍历的消息格式。一旦我们导航查阅域树，并定位了一个域的信息的一台权威 DNS 服务器，我们如何实际得到查询信息（该信息是我们为一个特定目的或应用，正在寻找的信息）呢？

与给定域关联的资源记录，提供了将问题映射到一个答案的方法。资源记录类型定义期望的结果类型，例如 A 资源记录类型将提供一个 IPv4 地址作为答案，而 AAAA 类型将提供一个 IPv6 地址。答案可以是“最终答案”或可通过另外的查询或其他方法得到期望答案所使用的信息。

㊀ 一个长时间存活的查询是这样一种机制，一个解析器请求接收区域信息变化的通知；有些像用于客户端的一条 DNSNOTIFY。

㊁ 更新租赁寿命机制，使一台 DHCP 服务器能够在一条 DNS 更新消息内，针对新的和刷新的租赁，将以秒为单位的对应客户端的租赁时间长度通知该 DNS 服务器。

第10章 DNS应用和资源记录

10.1 引言

本质上来说，DNS是将一个给定的信息片“翻译”为另一个相关的信息片。这个解析过程恰是DNS发明的原因，并已经被扩展，远远超出将主机名解析为IP地址这单项功能，反过来支持较宽种类的应用。几乎可以这样说，要求将一种形式的信息翻译为另一种形式信息的任意服务或应用，均可利用DNS。

在DNS中配置的每条资源记录使这项查找功能成为可能，它返回一条给定查询的一个解析答案。DNS服务器分析DNS消息[㊀]问题节的查询，针对该查询的QNAME、QCLASS和QTYPE，在相应域的区域文件内寻找一条匹配。每条资源记录有一个名字（即属主）字段、类（如果没有指明，则假定为因特网类）和类型字段。RData字段包含查询的相应答案。资源记录类型定义了问题的类型和格式（属主/名字字段）和对应答案（RData字段）。在一些实例中，多个资源记录可匹配被查询的名字、类型和类。在这些情形中，称为一个资源记录集合（RRSet）的所有匹配记录，在响应消息的答案节中返回。

多数新应用（但不是所有的）都要求新的资源记录类型，以便定义应用特定的信息，这些新的资源记录类型是通过IETF RFC过程来标准化的。本章描述了DNS中存储的各种形式的信息以及它们所支持的应用。在本章末尾，提供一个资源记录汇总，以便参考。

10.1.1 资源记录格式

首先让我们回顾一下一条资源记录的格式。当对查找信息的一条查询做出响应时，一台DNS服务器将资源记录信息放置在一条DNS消息的答案节之中。由DNS协议规定的“传输（on-the-wire）格式”，在DNS消息答案节的格式上下文中的图9-13做了介绍，出于方便，在图10-1中重新给出。

当在区域文件中表示资源记录时，所有这些字段（除了RDLength字段外）都被输入，当将资源记录信息放置在一条DNS消息中时，DNS服务器将RDLength字段插入。一般而言，一条资源记录的文本表示遵循如下所示的一种通用惯例。多数资源记录采用如下通用字段加以定义，虽然许多字段在RData字段内都有子字段，我们将在本章后面看到具体情形。

属主	存活时间	类	类型	RData

㊀ 参见图9-12。

1）属主（名字）。这个字段匹配正被查询的信息。

2）存活时间（TTL）。对于缓存这个信息的服务器和解析器而言，在这条资源记录内包含该信息的有效秒数。在 TTL 超期之后，资源记录信息必须从名字服务器和解析器缓存中被清除。可在每条资源记录基础上，指派确定 TTL，或在被略去的情况下，使用一个区域层次默认的 TTL 值（ $ TTL）。

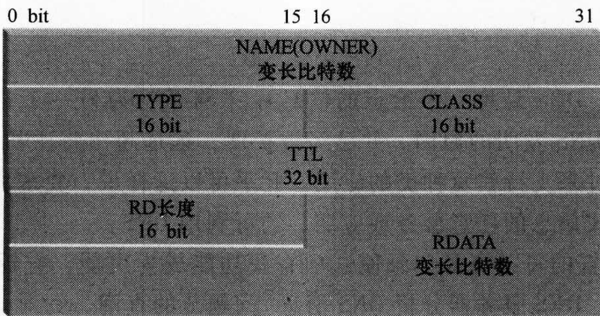


图 10-1　DNS 资源记录传输格式[99]

3）类。资源记录的类，对于因特网而言，通常是 IN。

4）类型。对应于正被查找信息的类型的资源记录的类型。

5）RData。通过匹配属主（名字）、类和类型字段内容，得到的对应于正被查找信息的“记录数据”或答案部分。

既然我们已经讲解了基本格式，则就准备好跳到支持它们的特定应用和资源记录。当我们回顾这些资源记录类型时，将回顾每个类型的解释，并给出一个例子。我们将讲解已经被 IETF“官方”接受的那些资源记录类型，即它们已经以一个 RFC 的形式发布；但是，发布为一个 RFC，并不保证在所有解析器和服务器间对该资源记录类型的统一实现。我们将指出，其中一些资源记录类型可能是新的或试验型的，而另一些则早已存在了数年时间。

对于我们将在本章讨论的每个记录类型，使用一种通用格式，显示资源记录字段和范例。对于每个记录类型，第一行或表头，都指派规范了如上定义的基本字段。第二行显示存在疑问的特定类型的这些基本字段的解释。给定类型的一个范例资源记录显示在第三行，可选的后续行汇总如下。

资源记录字段
资源记录字段数据类型
样例资源记录(可能有多项)

注意，术语“域名”是指一个 DNS 域的名字，术语“主机域名”是指一台主机的 DNS 名字。主机域名可采用区域文件定义，定义为完全合格的（FQDN）或简单的一个主机名（在“当前域”的上下文中进行解释）。当前域是在 named. conf 文件的区域声明中定义的，除非在区域文件中使用一条 $ ORIGIN 语句做出变更。

10.2　名字-地址查询应用

10.2.1　主机名和 IP 地址解析

首先，最常见的 DNS 应用是主机名解析，即查找一个主机域名并得到其对应的 IP 地址。支持两个资源记录类型，用于 IP 地址查询，一个资源记录类型用于 IPv4 地址，另一个用于 IPv6 地址。相应的反向记录利用 IPv4 和 IPv6 的一个通用记录类型，即指针（PTR）记录类型。

当管理一个混合的 IPv4-IPv6 网络时，注意 DNS 将强烈地影响使用哪个协议到达一台给定的目的主机。例如，如果我希望访问一个网站，我的解析器首先检索给定网站地址的一条 A 记录。在不能得到一个 IPv4 地址时，之后它会尝试一次 AAAA 记录查找，这次会成功。假定我的浏览器（TCP/IP 栈）支持 IPv6，那么连接将在 IPv6 上进行。不知道的是，我正在使用 IPv6。某些 IPv4-IPv6 迁移技术明显地强制 DNS 中的双协议查找（A 和 AAAA）。我们将在第 15 章讲解这些技术和 DNS 对 IPv4-IPv6 网络的整体影响。

（1）A-IPv4 地址记录。A 记录是一个通用的资源记录类型，用来将一个被查询的主机域名映射到一个 IPv4 地址。其格式遵循如下例子中的标准惯例。各主机可能有多个 A 记录，这可提供一个主机名到多个设备和/或接口的负载均衡或映射。

属主	TTL	类	类型	RData
主机域名	TTL	IN	A	IPv4 地址
www. ipamworldwide. com.	86400	IN	A	10. 100. 0. 99

（2）AAAA-IPv6 地址记录。基于一个主机域名的查找，AAAA（“四个 A”）记录提供一个 IPv6 地址。以类似于 A 记录针对主机名到 IPv4 地址查询的方式，进行格式化和处理，RData 字段包括一个 IPv6 地址，IPv6 地址可使用标准的 IPv6 缩略惯例进行缩略表示。

属主	TTL	类	类型	RData
主机域名	TTL	IN	AAAA	IPv6 地址
www. ipamworldwide. com.	86400	IN	AAAA	2001:DB8:3A::21:A450:1

（3）PTR——指针记录。PTR 资源记录提供从一个 IP 地址到一个 FQDN 的映射。PTR 记录用来映射 IPv4 和 IPv6 地址。PTR 的 IPv4 版本包括反向的 IP 地址并串接“in-addr. arpa.”作为属主字段，还包括对应的 FQDN 作为 RData 字段。IPv6 版本是如下形成的，以其十六进制冒号格式写出 IPv6 地址，包括所有的零，即填充前导零和双冒号简捷写法。之后丢掉冒号，将数字反向，之后串接“ip6. arpa.”。

在这个例子中的 IPv4 地址对应 10. 65. 32. 1，而 IPv6 地址是 2001：0DB8：0000：0000：0000：0000：0000：1001，或为缩略形式的 2001：DB8：：1001。

属主	TTL	类	类型	RData
以反向域格式表示的 IP 地址	TTL	IN	PTR	主机域名
1. 32. 65. 10. in-addr. arpa.	86400	IN	PTR	sf1. ipamworldwide. com.
1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0-. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 8. B. D. 0. 1. 0. 0. 2. ip6. arpa.	86400	IN	PTR	sf1. ipamworldwide. com.

10. 2. 2 别名主机和域名解析

CNAME 资源记录类型支持依据别名的一台主机域名的查找。CNAME 查找返回的不是一个 IP 地址，而是一台主机的域名，之后必须查询它的 IP 地址，但多数 DNS 服务器对一条 CNAME 的查询响应，将在 DNS 响应消息的附加节内包括对应的 A 和/或 AAAA 记录。同时，DNAME 记录提供查找域的一个类似别名功能。正如在前一章所讨论的，CNAME 和 DNAME 记录在处理非字节边界的反向域中是有用的。

(1) CNAME——规范的名字记录。CNAME 记录使主机别名的产生是可能的。属主字段包含要被查找的别名，RData 字段产生规范的主机域名。之后，将需要解析这个主机域名，从而得到主机对应的 A 和/或 AAAA 记录。

属主	TTL	类	类型	RData
产生主机域名别名	TTL	IN	CNAME	规范的主机域名
W3. ipamww. com.	86400	IN	CNAME	www. ipamww. com.

注意，为了将记录链接起来，配置一个 CNAME RData 字段指向另一个 CNAME 属主字段是不合法的。这个 RData 字段必须直接指向一个 A/AAAA 资源记录属主名。每个 CNAME 记录的属主名也必须是唯一的；单一别名不能解析为多个答案。如我们在第 9 章讨论的映射反向域名，对于这项功能，CNAME 记录证明是有作用的。

(2) DNAME——域别名记录。在 RFC 2672[107] 中定义的 DNAME 资源记录，支持将域名空间的整个子树映射到另一个域。开发 DNAME 记录的主要动力是，简化 IP 网络重新编址对 DNS 的影响。例如，如果运行 ipamww. com 域的公司被 acquired. com 收购，则 ipamww. com 名字空间可想象地被“移动”到 acquired. com 之下，方法是添加一个 DNAME 记录，如下所示。

属主	TTL	类	类型	RData
域名别名	TTL	IN	DNAME	目标域名
ipamww. com.	86400	IN	DNAME	ipamww. acquired. com.

在 ipamww. com 域树内查找主机的解析器，将被定向到 ipamww. acquired. com 域之下查找相同主机名。注意，这个场景要求 ipamww. com 的资源记录和子域要移植 (ported to) 到 acquired. com 域树内其对应的区域文件。RFC 2672 规定，DNAME 属主 (在这个情形中是 ipamww. com.) 区域一定不能有任何子域，也不能包含除 DNAME

和可能 CNAME 资源记录之外的任何资源记录。

10.2.3 网络服务定位

在一个网络上启动的 IP 设备，经常需要寻找特定的服务来进行设备初始化。DH-CP 通过指派某些选项值（例如 TFTP 服务器 IP 地址）的方法，提供了某个层次的服务定位，同时 DNS 使用服务定位资源记录类型（SRV）提供了一种服务定位机制。SRV 记录是一种非 DHCP 客户端或初始化后查找服务的客户端定位提供所请求服务的服务器的方法。

如果自 Windows 2000 出现以来，您都一直使用微软客户端和域控制器的话，那您可能非常熟悉 SRV 记录了。当 Windows 域控制器启动时，它为其 A 记录和 SRV 记录实施一次动态 DNS（DDNS）更新，使它们可有效地公告服务的可用性。通过在属主字段内使用下划线，也可容易地识别这些记录。对于一个给定域，SRV 记录属主是通过串接一项特定服务组成的，可通过一个特定协议（TCP 或 UDP）使用这些服务。为了消除与有效域名的冲突，加入了下划线。虽然从技术角度来说，依据 DNS RFC 1035，下划线不是一个有效的主机域名字符，但通过检查名字选项参数可配置微软服务器和 BIND 服务器，使其容忍下划线字符。虽然使用 SRV 记录是一个常见例子，SRV 记录当然不限于 Windows 应用，但迄今为止在 Windows 应用之外采用这种方法的做法还是有限的。

（1）SRV——服务定位记录。使用 SRV 记录的做法，使解析器客户端识别提供特定服务的服务器成为可能，这些服务如 LDAP、Kerberos 以及其他服务。在微软 Windows 客户端定位 Windows 域控制器的过程中，这个记录是至关重要的。

属主	TTL	类	类型	RData			
服务编码	TTL	IN	SRV	优先级	权重	端口	目标主机域名
_ldap. _tcp. ipamww. com.	86400	IN	SRV	10	0	389	ldap. ipamww. com.

属主字段由一个给定域的特定服务串接组成，可通过一个特定协议（TCP 或 UDP）使用这项服务。RData 字段包括一个优先级字段，当返回多个 SRV 记录时，该字段指令客户端使用具有数值较低优先级的目标（SRV 记录）。

使用权重字段来进一步将有相同优先级的记录作出优先区分。端口是 TCP 或 UDP 端口号，用来访问给定服务；访问目标（target）是运行具体服务之服务器的主机域名。

如果 DNS 服务器也不作为附加信息返回，则客户端可请求主机相应的 A 或 AAAA 记录，将主机指派为目标，目的是完成解析过程。一些例子如下：

_ ldap. _ tcp. ipamww. com. 86400 IN SRV 10 5 389 ldapeast1. ipamww. com.

_ ldap. _ tcp. ipamww. com. 86400 IN SRV 10 10 389 ldapeast2. ipamww. com.

_ ldap. _ tcp. ipamww. com. 86400 IN SRV 20 1 389 ldapeast3. ipamww. com.

在上述三个样例 SRV 记录中，将首先使用第二条记录。它与第一条记录具有相同的最低优先级数字（10），但它的优先级字段（10）比第一条记录（5）高，第三

条记录将最后被使用，原因是它虽然有低的权重，但却有较大的优先级数值。

在上述例子中的端口号 389 是给定服务的 TCP 或 UDP 端口号，目标（target）是运行所指派服务的服务器的主机名。如果也不作为从 DNS 服务器发出消息的附加节返回，则客户端要请求主机相应的 A 或 AAAA 记录，将主机指派为对应的目标，目的是完成解析过程。从语义角度而言，不需要在后两条记录中列出属主字段（假定它们是在区域文件内按顺序列出的），但我们却列出了属主字段，目的是强调针对一条_ ldap. _ tcp. ipamworldwide. com 的 SRV 查询，会返回这三条记录。

（2）AFSDB——DCE 或 AFS 服务器记录（试验型的）。在 RFC 1183[108] 中定义了 AFSDB 记录，其目的是支持一台服务器的定位，特别是 AFS（这是 Transarc 公司的一个注册商标，最初是 Andrew File System（安德鲁文件系统）的缩写）和开放软件基金会的分布式计算环境（DCE）的服务定位。

属主	TTL	类	类型	RData
单元域名	TTL	IN	AFSDB	子类 主机域名
ipamworldwide. com.	86400	IN	AFSDB	1 afsdb1. ipamworldwide. com.

RData 字段组成如下。

1）子类型字段，它识别单元域名（子类型 =1）的 AFS 3.0 卷位置服务器或给定单元域名的 DCE 目录服务的服务器（子类型 =2）。

2）主机域名字段识别服务器主机名。

AFSDB 资源记录没有被广泛使用，原因是该 SRV 资源记录类型在 DNS 内提供通用的服务器定位功能，事实上，RFC 5864[189] 规范了用于 AFS 的 SRV 记录使用情况。

（3）WKS——周知的服务记录（历史型的）。这个资源记录类型识别周知的服务，例如 FTP、telnet 以及其他服务，它们是在一个特定 IP 地址上可用的，为一台主机使用一个特定的协议（TCP 或 UDP）。这个记录没有被普遍使用，原因是 SRV 记录提供了类型的功能。

属主	TTL	类	类型	RData
主机域名	TTL	IN	WKS	IPv4 地址 协议 服务
server. ipamww. com.	86400	IN	WKS	10. 0. 199. 35 TCP SMTP FTP

10.2.4 主机和文本信息查找

TXT 记录是承载（workhorse）资源记录类型之一，经常用作一个中间（interim）资源记录，支持特定应用正在（pending）标准化的过程和实现。TXT 记录支持一个通用索引名的查找，例如一个域名、主机域名或其他属主值，并返回任意的文本信息。最近，TXT 记录已用作 DDNS 更新唯一性检查（现在是 DHCID 记录类型）和降低垃圾邮件的应用（SPF 记录类型）的中间阶段支持，这两者在本章后面讲解。

（1）TXT——文本记录。文本记录支持将多达 255B 的任意二进制数据与一个资

源记录的关联。在提供新服务的中间阶段支持方面，它已被证明是有多种功能的（very versatile）。

属主	TTL	类	类型	RData
索引名	TTL	IN	TXT	任意文本数据
Txt. cfo. ipamww. com.	86400	IN	TXT	“CFO Office(610)555-1212”

（2）HINFO——主机信息记录。HINFO 资源记录的 RData 字段支持对一台主机的处理器和操作系统的查找。

属主	TTL	类	类型	RData	
主机域名	TTL	IN	HINFO	CPU	操作系统
sf1. ipamww. com.	86400	IN	HINFO	VAX 770/11	UNIX

（3）HIP——主机身份协议记录（试验型的）。HIP 资源记录类型，支持试验型的主机身份协议（HIP），它本质上是将一个主机名与一个 IP 地址的关联进行了抽象，方法是在解析过程中插入了一个“主机身份”层。这使一个域名与一个主机身份的关联成为可能，之后主机身份与一个或多个 IP 地址关联。一个应用或上层协议可通过该 HIP 资源记录查找一台主机，并得到主机标识符（以一个公开密钥的形式表示）和其他主机身份信息，包括主机或一台汇聚服务器的 IP 地址，通过汇聚服务器可连接到移动设备。

属主	TTL	类	类型	RData					
主机域名	TTL	IN	HIP	HIT Len.	PK Alg	PK Len.	HIT	公开密钥	RVS
Hiphost. ipamww. com.	86400	IN	HIP	16	2	24	Iil…	8L9d…	rs. ipamww. com

Rdata 字段定义如下。

1）HIT Len（HIT 长度）。以字节表示的主机身份标签（HIT）的长度；这个字段是由服务器插入的，目的是线路（wire）传输，在一个区域文件内是不显示的。

2）PK Alg（PK 算法）。用于产生公开密钥的算法

① 0 = 不存在密钥。

② 1 = DSA 格式的密钥。

③ 2 = RSA 格式的密钥。

3）PK Len（PK 长度）。以字节为单位表示的公开密钥长度；这个字段是由服务器插入的，目的是线路（wire）传输，在一个区域文件内是不显示的。

4）HIT。主机身份标签，这是主机标识符的 128bit 散列值。

5）公开密钥。与主机关联的公开密钥，可用于验证来自主机的签名消息。

6）RVS（可选的）。一个或多个汇聚服务器主机域名，用于连接到移动设备。

（4）RP——负责人记录。RP 资源记录使一个电子邮件地址和其他文本信息与域树中的一个节点（是一台端主机或域）关联成为可能。RData 字段包含一个电子邮件地址，其格式为不带@号的形式；相反，一个点号（.）替代了@号。RData 字段的

第二个字段指明这样一个记录，它用于附加文本信息，可作为一个附加查询的答案。

属主	TTL	类	类型	RData
主机域名	TTL	IN	RP	电子邮件地址　TXT 指针
payroll. ipamww. com.	86400	IN	RP	cfo. ipamww. com. cfo-contactinfo. ipamww. com.

在上面的这个例子中，我们使用一条 RP 记录，将支付服务器与我们的 CFO 关联起来，通过 cfo@ipamww. com 可找到他（在电子邮件地址字段中以“.”替换“@”）。TXT 指针字段指向包含附加信息的一个资源记录，例如下例：

cfo-contactinfo. ipamww. com. 86400 IN TXT “CFO Office (610)-555-1212”。

10. 2. 5 DNS 协议运营性的记录类型

两个“管理型的”资源记录类型，使区域权威信息（SOA 记录）的规范确定成为可能，还有使这个域及子域（NS）的名字服务器的委派成为可能。对于保持一个区域内 DNS 数据同步以及保持委派链实际上是沿域树向下的关系，要使这两方面高效可操作，这两个记录类型是有指导作用的（instrumental）。

（1）SOA——权威起始记录。对每个区域，仅可有一个 SOA 记录，如果存在初始默认 TTL（$ TTL）语句，则在区域文件中后跟该语句。SOA 记录定义了这个区域的权威域名，还有附加的区域维护信息。SOA 记录由如下字段组成。

属主	TTL	类	类型	RData
域名	TTL	IN	SOA	Mname 联系信息　序列号　刷新间隔　重试间隔　超时间隔　负面缓存
ipamww. com.	86400	IN	SOA	ns1. ipamww. com admin. ipamww. com. 3945 2h 30m 1w 1d

1）这个区域文件的域名，包含权威信息。

2）TTL，存活时间。

3）记录类（对于因特网是 IN）。

4）记录类型（SOA）。

5）主 DNS 服务器名（MNAME）。DNS 服务器的名字，它是这个域（区域）的主服务器。

6）域联系人的邮件地址（将“@”替换为“.”，从而 admin@ipamworldwide. com 写为 admin. ipamworldwide. com.）。注意在@号之前带有点号的电子邮件地址，应该以一个斜线为前缀。因此 super. admin@ipamww. com 将被编码为 super\. admin. ipamww. com。

7）区域的序列号。在每对区域数据作出一次改变时，就增加 1——这使从属服务器能够识别对区域数据所作的改变。

8）刷新间隔。从属服务器查询主服务器，请求区域更新的时间周期。

9）重试时间。如果不能联系到主服务器，则从属服务器将等待这段时间量，再

重试联系主服务器。

10）超期时间。如果在这段超期时间量之后，不能联系到主服务器，则从属服务器将删除区域信息，并不再认为它自己是权威的，由此对此区域的权威性就过期了。

11）负面的缓存 TTL。维护来自其他服务器的负面响应的缓存时间段，例如一个指定的域或记录不存在。

我们的 ipamww. com 区域文件的一条范例 SOA 记录，也许看起来有点像 ipamww. com. IN SOA dns1. ipamww. com dnsadmin. ipamww. com (

1 ; serial number

2h ; refresh interval of 2 hours

30m ; retry after 30 minutes

1w ; expire after 1 week

1d) ; negative caching TTL of 1 day

（2）NS——名字服务器记录。NS 记录支持对一给定区域的权威名字服务器的查找。对于 NS 数据库的分发而言，NS 记录是关键。在将一个子域委派给另一个管理权威的过程中，出于冗余性考虑，子域管理员必须运行至少两台名字服务器。在遍历域树时，这些 NS 记录使域树中沿解析路径被查询的名字服务器，能够以指向沿树向下的另一台名字服务器的索引（referral）作出应答，该名字服务器拥有拟查找目的地的更多信息。每个区域也必须针对其权威名字服务器，至少声明两条 NS 记录。

属主	TTL	类	类型	RData
域名	TTL	IN	NS	名字服务器域名
ipamworldwide. com.	86400	IN	NS	ns1. ipamworldwide. com

注意在 RData 字段中的名字服务器主机名，应该有一条相应的 A 或 AAAA 记录，以便完成到一个可达 IP 地址的必备解析。这被称作一条“黏结”记录，原因是它将所期望域的权威名字服务器主机名的解析与那台名字服务器的 IP 地址“黏结”在一起。

10.2.6　动态 DNS 更新唯一性验证

DHCID——动态主机配置识别符记录。动态 DNS 支持 DHCP 客户端的指派 IP 地址信息的 DNS 信息更新。因此，代表客户端的一台 DHCP 服务器，或客户端自己，都能够以客户端的 IP 地址或主机名关联，通过 A/AAAA 和 PTR[⊖] 记录，更新 DNS。非常有可能出现如下情形，即同一主机名/FQDN 可能为多个 DHCP 客户端所声明，或一个客户端声明一个主机名，但该主机名已经被指派给一个预先确定的（例如，静态分配地址的情形）设备。

⊖ 在 DHCID RFC 中，将客户端识别信息与 PTR 记录相关联，当前是没有规范的。

DHCP记录提供在DNS中的客户端识别信息，以此唯一地将特定DHCP客户端与主机名/FQDN（正由该DHCP服务器更新的）相关联。DHCID记录将在DNS更新消息的前提条件节中定义，目的是验证更新的记录“属主”。请参看前一章的DNS更新节，了解更多细节以及这个前提条件处理的一个例子。

DHCID记录使用相应A或AAAA记录的相同的属主字段。该记录的RData部分是这样形成的，即在如下串接字段之上使用SHA-256算法，实施一个单向安全散列函数。

1）识别符类型码（2B）。识别生成这个散列值中使用的在DHCP报文内部的信息。可能信息包括客户端硬件地址、客户端识别符选项或设备唯一识别符（DUID）。

2）摘要类型码（1B）。识别散列算法。RFC定义了数值0（保留）或1（SHA-256），但IANA维护了未来数值指派的一个注册机制（registry）。

3）来自DHCP报文的数据的摘要，是由识别符值与客户端的FQDN串接加以识别的。

属主	TTL	类	类型	RData
主机域名	TTL	IN	DHCID	识别符类型　摘要类型 {识别符类型,fqdn} 的 SHA-256 散列值
w3. ipamww. com.	86400	IN	DHCID	A1B87Y2/AuCcg8e93aQcjl...

10.2.7 电话号码解析

DNS被证明功能是非常多样的，甚至可被用来将电话号码映射为IP地址，这对于VoIP应用或IP应用上的相关电话功能是有用的。已经定义了ENUM（E.164电话号码映射）服务来支持这样的解析。ENUM以ITU E.164格式支持将电话号码映射为统一的资源识别符（URI）[⊖]。这个映射主要是由命名权威指针（NAPTR）资源记录类型来实施。

注意，多数企业IP PBX系统提供它们自己的目录，将内部PBX电话号码映射到目的电话的IP地址，所以在这样的环境中ENUM通常是不能实现的。但是，VoIP提供商具有既得利益，最大限度地确保呼叫保持在他们的或他们合作伙伴的IP网络和接入网络上，以便降低支付给非伙伴网络提供商（或糟糕的情况下，支付给竞争对手）的呼叫处理成本。ENUM是依据电话号码映射或解析而支持这种呼叫路由的关键。那并不是说，您不能在企业网络内看到ENUM的使用。ENUM采用优先级设置，提供到多个目的地的解析，在可达性或联系管理类型应用中它可找到应用用途。

正如刚刚提到的，NAPTR资源记录提供了将电话号码信息翻译为目的统一资源识别符的功能。当前在RFC 3403[109]中定义，NAPTR记录最初是用来定义提供如下功能的，即为动态委派发现系统（DDDS）提供迭代地将任意一个字符串解析为一个URI的功能。在RFC 3402[110]中提供了有关DDDS的一些背景信息，但它最初源于这样的期望，即定义一个解析过程，它可输入一个资源名（例如一个特定应用或特定数

⊖ 一个URI是一个因特网识别符，由一个统一的资源名（URN）和一个统一的资源定位符（URL）组成。一个简单例子：对于URL http://ipamworldwide. com和URN file. txt，对应的URI是http://ipamworldwide. com/file. txt。

据片），在其中没有包含网络位置信息，并将其解析到一个目的资源识别符，方法是对一个数据库施用一系列迭代规则。资源名的规范与定位或解析它的过程的这种分离方法，有利于作出改变和资源的重新委派，而不影响端用户应用的命名惯例。

这项工作扩展超出了解析资源名的范畴，扩展到支持通用查找字符串的解析，并演化成 DDDS，它使用 DNS 作为规则数据库的一种形式。NAPTR 记录支持在 DNS 内部指定这样的规则，有时要使用多条 NAPTR 记录来完整地完成解析过程。每条 NAPTR 记录将一个给定的项字符串（即一个有效的 DNS 域名）转换为一条规则，该规则可施用到该字符串，从而推导出要查找的下一个字符串。直到到达一条终结规则，以及最终结果被返回到发出请求的应用，这个过程的迭代才终止。

对于在 IP 上提供语音服务的服务提供商而言，NAPTR 记录是 E. 164 电话号码映射服务器的构造块。RFC 3761[111] 为 ENUM 应用提供了 NAPTR 字段的“应用特定的”解释。一条 NAPTR 记录可被用来查找一个目的电话号码，并将该号码解析为一个目的地，例如一台会话初始协议（SIP）服务器、电子邮件地址或其他 URI 格式的目的地。NAPTR 记录也支持定义正则表达式的能力，它提供逻辑规则，作为解析其定位拟到目的地过程的“下一步”。

E. 164 是格式化电话号码的一个国际电信联盟（ITU）标准。“完全合格的”电话号码，意味着它们是全球唯一的，给定国家码前缀后跟一个国家特定的电话号码格式，被表示为带有一个加号前缀，例如 +1 - 610 - 555 - 1234。这很像 IP 的反向域情形，将一个电话号码格式化，要求从左到右读取资源记录的一个类似惯例，这是从比较具体到不太具体的过程。这个惯例要求将完全合格的电话号码（去掉加号）反向，并将每个数字以“点”号（.）分隔。

主要 . arpa 顶层域的使用。类似于 ip6. arpa 和 in-addr. arpa. 域结构，e-164. arpa 域是一个“反向”域，其中它支持一个带结构数字值的查找，即一个电话号码。像其他 . arpa 查找一样，域结构也是从底向上组织的，从一般到特殊，或国家代码到电话线号码的顺序。因此，完全格式化的 E. 164 电话号码是反向的，每个数字采用点号分隔，并附加 e164. arpa. 域后缀，如图 10-2 所示。

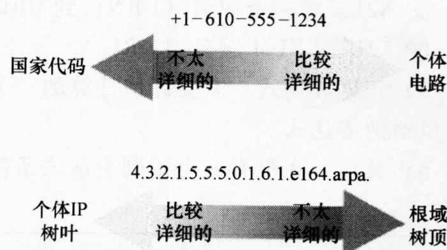


图 10-2 映射到域结构的电话号码

这个结构恰如其分地导入到电话号码空间的分段。例如，域 1. e163. arpa 指代所有国家代码为 1 的电话号码，并可被委派给这样的一个号码权威机构。类似地，44. e164. arpa. 可被委派给英国电话号码权威机构。在每个这样的域内，可依据国家的编号计划，完成进一步的委派。例如，在美国国内，一个区域代码代表下一个逻辑的管理委派点，后面串接交换机（exchange）。因此，1. e164. arpa 可将 0. 1. 6. 1. e164. arpa. 区域委派给 610 区域码的编号管理员，他接下来将 5. 5. 5. 0. 1. 6. 1. e164. arpa. 委派给 610 区域码内负责 555 交换机的那些管理员。

（1）NAPTR——命名权威点记录。NAPTR 记录提供将一个字符串㊀或电话号码信息转换为目的统一资源识别符的方法。NAPTR 记录利用上面在其自己字段内描述的 e164. arpa. 域命名惯例，用作电话号码的查找格式。不幸的是，迄今为止这是唯一容易的部分。NAPTR 记录在其 RData 字段内包含许多附加的子字段。下面描述附加的子字段，为 NAPTR 记录的 ENUM 应用提供了多个例子。

1）顺序（Order）字段。指定 RRSet 内多条记录被处理的顺序；首先处理低编号顺序的记录。

2）优先级字段。指定带有相等“顺序”值的记录被处理的顺序。首先处理低编号优先级的记录。

3）标志。就解析过程中“下一查找”提供有关信息。迄今为止，已经定义了四个标志数值，但标志字段可以是空的。

①“u”这条记录的正则表达式的输出是一个统一的资源识别符，即这是一个终结解析。

②“s”下一条查找应该是针对 SRV 记录的。

③“a”下一条查找应该是针对 A、A6 或 AAAA 记录的。

④“p”依据服务字段中确定的协议，下一个查找是特定于协议的。

4）服务。这个字段对服务编码，这些服务基于存在疑问的应用，是可用的。这个字段包括所提供解析的类型、一个“+”号或冒号，后跟协议值，例如 http、sip、mailto、ftp、tel，还有其他的协议，这里仅列举这些㊁。类型或解析的例子包括以下内容。

① I2L。URI 到 URL。

② N2L。统一资源名（URN）到 URL。

③ E2U。ENUM 服务到 URI。

5）正则表达式。需要评估计算的一个编码过的表达式。这个字段的语法是一个 sed 风格的表达式。

6）替代。在没有一个正则表达式条件下，一个替代的“下一查找”完全合格的域名。

属主	TTL	类	类型	RData
域名	TTL	IN	NAPTR	顺序　优先级　标志　服务　正则表达式　替代
me. ipamww. com	86400	IN	NAPTR	10 5 “s” “N2L + http” “ ” www. ipamww. com.
4. 3. 2. 1. 5. 5. 5. 0. 1. 6. 1. e164. arpa	86400	IN	NAPTR	10 20 “u” “E2U + sip” “! ^. _ $! sip:me@ipamww. com. !”

让我们更仔细地看看上面的两个范例 NAPTR 记录。第一个例子提供了

㊀ “字符串（复数）”指代文本或数据字符串（复数）。幸运的是，这不是 DNS 的“字符串理论”。

㊁ 拟了解 ENUM 目前指派的服务值，请参考 http://www. iana. org/assignments/enum-services。

me. ipamww. com 解析的一条规则。标志字段值“s”向解析器指明，下一查询应该是对 SRV 资源记录的一条查询。服务字段指明使用 HTTP 的一个 URN 到 URL 服务。因为正则表达式字段是空的，所以替代字段被用作解析处理的结果。

第二个例子突出了一个 ENUM 应用例，其中可解析对一个电话号码的查找。“u”标志指明所提供正则表达式的结果将是一个 URI，之后可将该 URI 解析到一个 IP 地址。服务字段指明使用 SIP 的 ENUM 服务。正则表达式字段由两个子字段组成，以“!”字符封装。第一个字段包含“^. * $”，被解释为“从行开始（^）起到行尾（$）实施匹配，零个或多个（*）字符（.）”，即匹配整个属主字段。正则表达式的第二部分包含“sip：me@ ipamworldwide. com”，这作为我们正则表达式的结果被返回。在这种情形中没有使用替代字段。

之后得到的 URI，sip：me@ ipamworldwide. com 将发起一条 DNS 查询，查询 ipamworldwide. com 的一个地址（A 或 AAAA）记录。注意，一些 DNS 服务器可返回相关 A 或 AAAA 记录，作为查询响应中的附加信息，它包含 NAPTR 记录。得到的 IP 地址将被用作目的地址，以便发起到“me”用户的 sip 会话。

10.3 EMAIL 和反垃圾邮件管理

垃圾电子邮件或没有被请求的块（unsolicited bulk）电子邮件，自因特网诞生伊始，一直就是无聊的废话，即使在早期日子里它也是经常不被赞同的一种做法。尽管如此，随着因特网的爆炸式增长，垃圾邮件的总量似乎增长得更快。存在各种技术来对抗垃圾邮件，多数这样的技术涉及 DNS 的使用。为了理解 DNS 如何能够帮助减少垃圾邮件，我们将首先剖析一下一个电子邮件的传输（包括在电子邮件交付中 DNS 的角色），之后回顾一下在各种反垃圾邮件解决方案中 DNS 的使用。

10.3.1 电子邮件和 DNS

典型情况下，一封电子邮件源于一个人，并被发送到一个或多个接收者。每个电子邮件地址的格式如 mailbox@ maildomain。通常情况下，mailbox（电子邮箱）指代人名或一个邮箱的主人或电子邮件账号，而 maildomain（邮件域）典型地是公司或因特网提供商名字，是交付到对应邮箱或邮件交换中心（exchanger）的目的域。电子邮件是用如下方法发送的，使用一个电子邮件客户端，例如微软 Outlook、Eudora 或基于 web 的客户端（例如 yahoo 和 google），当由源发信人发送时，客户端就连接到一个简单邮件传递协议（SMTP）服务器（使用 SMTP）发送电子邮件。就像电子邮件的默认路由器一样，SMTP 服务器负责将电子邮件转发到它的目的地。

为了消息传输，SMTP 服务器必须将邮件域（maildomain）解析到一个 IP 地址。自然地这是使用 DNS 对邮件交换器（MX）记录类型以及对应的 A 或 AAAA 记录类型的查找，完成这项工作的。

MX——邮件交换器记录。邮件交换器记录被用来定位一个特定域的一台电子邮件服务器或多台服务器。如果发送目的地为 tim@ ipamworldwide. com 的一封电子邮

件，SMTP 服务器将使用 DNS 寻找可为 ipamworldwide. com 域中各用户接收电子邮件的主机（可能是多台主机）。可为每个域构造一条以上的 MX 记录，每条记录定义时都带有一个不同的优先级数值。优先级字段的使用，使发送 SMTP 服务器能够对目的主机（将为给定域转发电子邮件）进行优先级排序，如果不可用，就转向第二个（或第三个等）可选目的地。优先级数值越低，则所列目的地被首选的可能就越高。在下面的例子中，对 ipamworldwide. com 域，有两条 MX 记录。目的 smtp1（低优先级数值）要比 smtp2 优先选中。但是，如果 smtp1 不可用，则这种机制提供电子交付的一台备份服务器。

属主	TTL	类	类型	RData
电子邮件目的域	TTL	IN	MX	优选　邮件服务器主机域名
ipamworldwide. com.	86400	IN	MX	10 smtp1. ipamworldwide. com.
ipamworldwide. com.	86400	IN	MX	20 smtp2. ipamworldwide. com.

注意，为了完成所需的一个可达 IP 地址的解析，在 RData 字段内的邮件服务器主机域名必须有一条对应的 A 或 AAAA 记录。许多 DNS 服务器在 MX 查询响应的附加节内提供这些地址记录。

在解析目的邮件服务器时，SMTP 服务器使用 SMTP 向目的地发送该消息。最终的目的服务器，接收者电子邮件客户端要连接到该服务器，它必须支持邮局协议（POP）或因特网消息存取协议（IMAP），以便使客户端可检索电子邮件消息。因此，当您的电子邮件客户端执行一条"send/receive"（发送/接收）操作时，它利用 SMTP 将外发消息发送到其配置的 SMTP 服务器，利用 POP 或 IMAP 从配置的 POP/IMAP 服务器（可能是多台）处检索收到的电子邮件消息。

图 10-3 突出显示了两台服务器间一个非常简单的 SMTP 事务（当我的朋友迈克发送给我一条电子邮件时）。在图的左侧，迈克使用他的电子邮件客户端，撰写一封给 tim@ ipamworldwide. com 的电子邮件，并发送该邮件。依据针对 ipamworldwide. com 的 MX 记录（可能是多条）解析结果，他所配置的 SMTP 服务器将该消息转发到目的服务器。他的 SMTP 服务器在端口 25 上发起到所解析目的服务器的一条 TCP 连接。

一旦建立 TCP 会话，则 SMTP 应用就利用该会话进行握手，并处理消息。消息的信封部分以 HELO（或 EHLO，这是增强的 HELO）开始，它携带发送实体的身份。MAIL FROM 语句指明消息的源，后跟 RCPT TO 语句，指明目的邮箱。在交换过程中的此点，如果目的邮箱未知或被阻止，或如果"来源地址"（from address）是被禁止的，则接收者服务器可拒绝接收消息，并关闭连接。否则，事务继续进行，接着传递数据或消息部分㊀。接收邮件交换器存储电子邮件消息或将其转发到目的邮箱所在的服务器。

接收电子邮件服务器所使用的存储转发方法，也可由中间电子邮件网关（即消

㊀ 注意一封电子邮件的消息部分是由一个头部和体组成的。作为一个参考文献，RFC 2821[163] 定义 SMTP 规范，而 RFC 2822[164] 定义了电子邮件的因特网消息格式，它定义有效的头和数据语法。

息传递代理）使用，用来提供多跳电子邮件交付。如上所述，一个目的邮件域到多个 MX 记录的解析，意味着识别一个“目的”邮件服务器的这种能力，该服务器可能是或可能不是最终目的地，由此处预期中的接收者检索电子邮件。MX 记录优先级字段提供了对进入邮件服务器或网关相对优先级的控制，同时提供了基于可用性和性能而在多个选项间选择的能力。

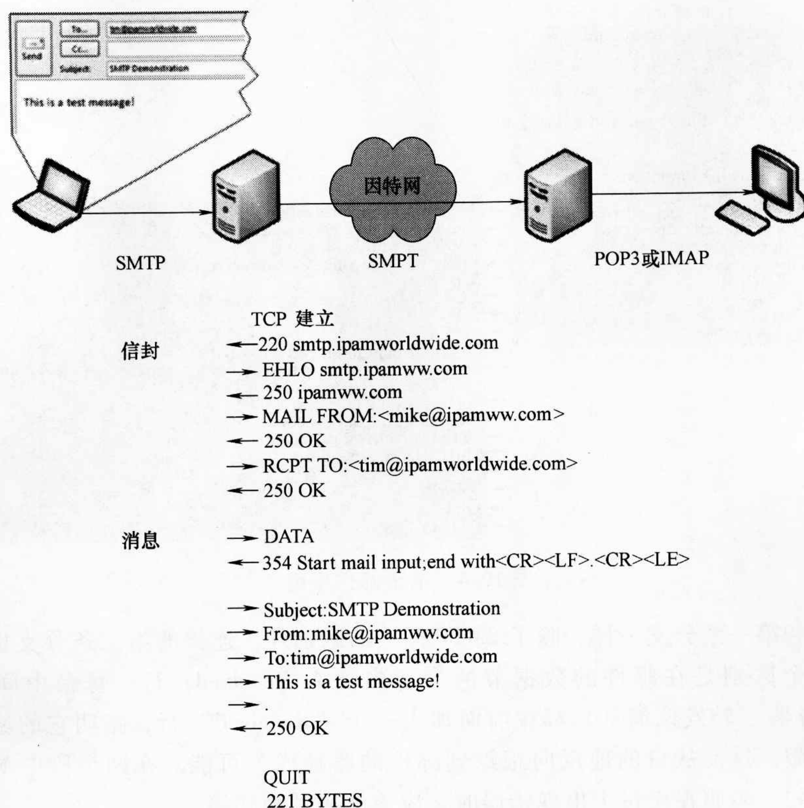


图 10-3　简单 SMTP 事务例

图 10-4 形象地说明了使用 SMTP 的一个两步骤电子邮件交付的场景。在这个场景中，我正在发送如图 10-3 所示的同一电子邮件。但是，在这种情形中，也许预期中的目的服务器 smtp. ipamworldwide. com 正忙，并拒绝一条直接的连接。因为通过一条 DNS MX 查询解析了 ipamworldwide. com 服务器和一个 mta-gateway. com 服务器，所以我们的外发邮件服务器将尝试向第二项选择 mta-gateway. com 发送该电子邮件。

在接收来自我的邮件服务器的 SMTP 传输过程中，mta-gateway. com 服务器实际上同意了代表我将该邮件转发到最终目的地。在尝试第二条传输路径之前，我的邮件服务器和 mta-gateway. com 服务器之间的事务就完成了。SMTP 使用一种存储转发方法，而不是每条消息的同步中继。

除了 SMTP 服务器的区别之外，传输的第一条分支看起来非常类似于图 10-3 中

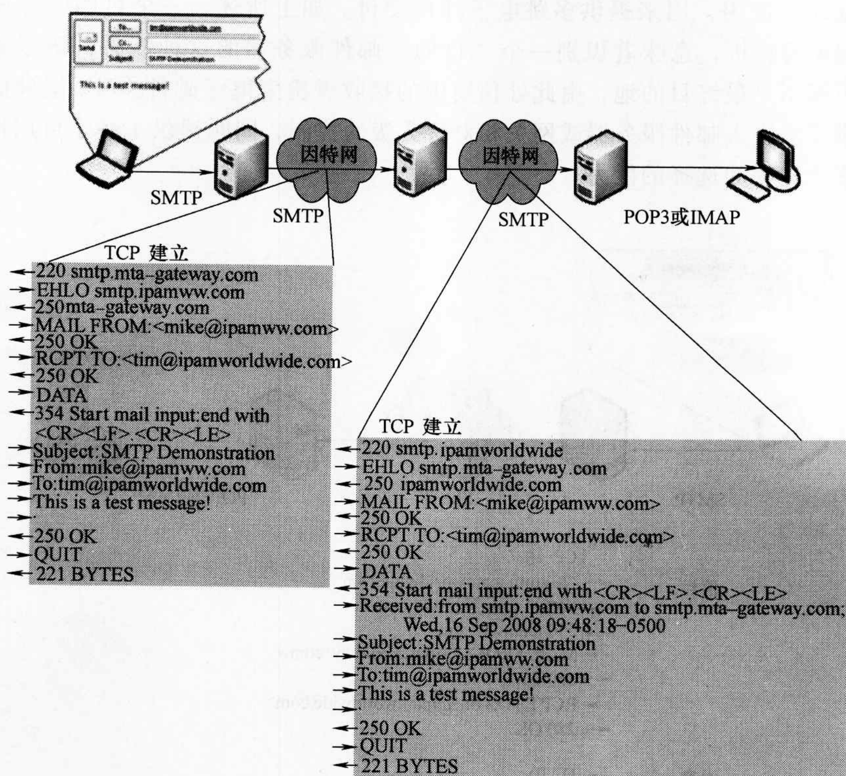


图 10-4 电子邮件中继

的路径。和第一条分支一样，除了 SMTP 端点的区别外，连接的第二条分支也是类似的。另一个区别是在邮件的数据节的头部内插入 Received：行。每台中间经过的 SMTP 服务器（转发该消息）都在前面加入一个“Received”行，指明它的域名和相应的时间戳。这使从目的地反向跟踪到邮件的路径成为可能。在两个段中 RCPT TO 行保持相同，表明在交付中出现错误时，应该发送到的邮箱。

一封电子邮件的消息部分由一个头部和体组成。每个头部字段由一个英文字后跟一个冒号和一个值组成。头部包含各种数据，包括如下字段：

1）源发字段。From、sender、reply-to、orig-date。

2）目的字段。to、cc、bcc。

3）识别字段。Message-id、in-reply-to、referencces（参见）、msg-id、id-left、id-right、no-fold-quote、no-fold-literal。

4）信息型字段。Subject（主题）、comments（注释）、keywords（关键词）。

5）重发字段（例如由一项电子邮件服务将一条消息重新引入[㊀]因特网有关的信息型字段）。Resent-date、resent-from、resent-sender、resent-to、resent-cc、resent-bcc、

㊀ 重新引入不是转发。带有原始发送者（而不是传输者）信息的一封电子邮件的传输被认为是重新引入。转发使用执行转发的邮箱作为发送者。

resent-msg-id。

6）源跟踪信息。Trace、return、path received、name-val-list、name-valpair、item-name、item-value。

我们已经总结了基本的电子邮件收发过程和信息类型，这些可被包括在一条给定的电子邮件消息内，原因是在验证发送者是电子邮件的一名合法或可接受的发送者时，不同的反垃圾技术利用不同的信息源。

10.3.2 白名单或黑名单方法

白名单或黑名单的使用[190]，为接收者的电子邮件服务器通过 DNS 查找一名发送者的 IP 地址，并验证其合法性，提供了一种简单方法。典型情况下，这种查找是如下形成的，即将电子邮件消息的源 IP 地址反向处理，就像形成 PTR 记录时所做的那样。注意正在被分析的源 IP 地址是电子邮件被直接接收处的 IP 地址，也许是一个电子邮件网关，它可能是或可能不是原始发送者。但是，这种列名单的意图，是依据 IP 地址来识别电子邮件的这种发送者是否为合法的。

在这个场景中，在反向的 IP 地址后附加一个给定的域名，典型情况下是黑名单提供商的域名。由此，通过串接形成这个“主机域名”，使用 A 字样记录查询类型（不是 PTR），在 DNS 中查询。基于是否找到该记录（在这种情形下经常返回 128/8 地址块内的一个 IP 地址）和列表是否发布已知的垃圾发送者（黑名单或阻塞名单）或已知的非垃圾发送者（白名单），对查询答案进行解释。

例如，在接收到带有源 IP 地址为 192.0.2.95 的一封电子邮件消息时，我的电子邮件服务器形成对主机名 95.2.0.192.spamblacklist.org 的一条 A 记录查询，其中假定我所选中的黑名单提供商在 spamblacklist.org 域内发布查找。在接收到带有答案（IP 地址）127.0.0.5 的一条应答时，我的电子邮件服务器将该电子邮件分类为垃圾邮件，并拒绝接收该邮件。另一方面，如果对查询返回 NXDOMAIN，则可允许接收该电子邮件。一项白名单列表服务，它发布已知的、名副其实的电子邮件服务器地址，将基于 DNS 查找，得到相反的解释。

10.3.3 发送者策略框架

发送者策略框架（SPF）目前由 RFC 4408[112]定义，该 RFC 处于试验状态。SPF 使一个组织机构发布其自己授权的外发电子邮件地址的列表、一个自发布的白名单，虽然本质上来说要更加复杂。在 SPF 下，所接收到的电子邮件消息的信封信息，是要检查的，来自接收者的一条 SPF DNS 查询的形成，要依据发送者、发送者的域以及源 IP 地址。在接收到一条电子邮件消息时，接收者电子邮件服务器将发起一条查询，查询源域名的一条 SPF 资源记录。SPF 记录是作为“机制”的一个字符串编码的，这些机制被用来处理源 IP 地址（电子邮件是从该地址发出的）、MAIL FROM 或 HELO 身份的域部分以及从 MAIL FROM 或 HELO 身份得到的发送者。

（1）SPF——发送者策略框架记录。发送者策略框架，尝试提供如下方面的验证，即配置哪些主机为一个给定的域发送电子邮件的验证。即 SPF 寻求从伪造域中去

除垃圾电子邮件。一台接收者电子邮件主机可查找发送者域的 SPF 记录，用来验证发送电子邮件的主机是否匹配发送者所授权的那些主机。SPF 版本 1（也称作 SPF 经典版）是在 RFC 4408 中描述记录的，它利用了 SPF 资源记录。域管理员可以配置电子邮件主机 DNS，这些主机映射到每台主机的 mailfrom 和 SMTP HELO 身份。SenderID 是一项相关的垃圾邮件检测技术，它也使用了 SPF 资源记录类型，但它分析来自一封进入电子邮件消息的不同信息。稍后我们将讲解 SenderID。

注意，由于在 IETF 发布 RFC 4408 之前，SPF 的各种实现都使用 TXT 记录，所以出于后向兼容目的，多数实现将同时使用 SPF 和 TXT 记录，但如果返回了 TXT 和 SPF 记录，则一个 SPF 兼容的解析器将丢弃 TXT 记录。SPF 记录的格式与 TXT 记录的格式是等价的。但是，SPF 应用了一种特定的语法，而不是任意的文本。该语法包括一个版本字符串（对于 SPF，v = spf1；对于接下来要讲解的 SenderID，v = spf2.0），后跟一个空格，接着是一项或多项，它们定义了有关资源记录类型或 IP 网络地址、修饰符甚至宏的标识符（qualifier）。

属主	TTL	类	类型	RData
域名	TTL	IN	SPF	版本、命令（directives）和/或修饰符
smtp. ipamww. com.	86400	IN	SPF	v = spf1 + ip4:192.0.2.32/30 - 所有的
smtp. ipamww. com.	86400	IN	SPF	spf2.0 pra + ip4:192.0.2.32/30 - 所有的

（2）各种机制。各种机制支持在 SPF（或 TXT）记录内部指定匹配准则，一台接收电子邮件服务器可查询该记录，以便验证一个给定电子邮件消息的发送者。在指定 SPF 版本（当前版本为 1，即“v = spf1”）之后，在 SPF 记录的 RData 字段内定义各种机制，是按照从左到右的顺序对各种机制进行评估的。如果基于对机制的评估，一种机制得以通过，则就通过了验证；否则，直到找到一种机制时通过或失败，或没有定义其他机制的情况出现之前，就都要测试下一个机制。

每种机制都是以一个标识符定义的，这是指令邮件或垃圾过滤器服务器如何接收一个给定“匹配”的一个前缀。

1） + =通过（默认的）。如果这种机制得以匹配，就认为这种机制通过了。

2） - =失败。如果这种机制得以匹配，则认为这种机制是一次失败。

3） ~ =软失败。如果这种机制得以匹配，就认为这种机制处于中性和失败之间的某个位置；如果这种机制得以匹配，则这种解释将不会使这项检查就失败了，而是保留它以便进行密切的检查。

4）? =中性的。如果这种机制得以匹配，就认为这种机制是中性的。

可与如下基于资源记录检查的机制一起，来使用标识符，以便定义一种给定机制的解释，如下各例所示。

1）a =查找源域（来自 MAIL FROM 或 HELO 身份）的 A 记录；如果它匹配消息的源 IP 地址，则这种机制就匹配了。这可将范围限制在一个特定域和/或在地址中要比较的 CIDR 比特数，如下各例所示。

① +a =通过，如果源域的 A 记录查询匹配源 IP 地址。

② －a：ipamworldwide. com＝失败，如果 ipamworldwide. com 的一条 A 记录查询匹配源 IP 地址。

③ ～a/24 软失败，如果通过源域的 A 记录查询，所检索的 IP 地址的前 24bit，匹配源 IP 地址的前 24bit。

2）mx＝查找源域（来自 MAIL FROM 或 HELO 身份）的 MX 记录；对于每个要解析的 MX 查找，查找相应的 A 记录；如果它匹配消息的源 IP 地址，则这种机制就得以通过。和 a 机制一样，mx 机制可将范围限制在一个特定域和/或在地址中要比较的 CIDR 比特数，如下各例所示。

① ＋mx：ipamworldwide. com/28＝通过，如果返回与一条 MX 记录查找相关联的一个 A 记录，其中前 28bit 匹配该消息的源 IP 地址的前 28bit。

3）ptr＝查找对应于电子邮件消息的源 IP 地址的 PTR 记录（可多达 10 条）；之后与 PTR 查找中返回的每个域名进行比较。

① 检查所返回的域名匹配电子邮件消息的源域。

② 检查相应 A 或 AAAA 记录所返回的一个 IP 地址，要匹配源 IP 地址。

如果上述两个条件成立，则这种机制得以通过。这种机制可进一步受到一个域名的限制，该域名可被用来过滤多个返回的 PTR 查找的域名，如下各例所示。

① －ptr：失败，如果在源 IP 地址的 PTR 查找过程中所返回的一个域名，匹配源域，以及如果在 PTR 查找过程中所返回域名对应的 A/AAAA 域名，匹配电子邮件的源 IP 地址。

② ＋ptr：ipamworldwide. com：通过，如果在源 IP 地址的 PTR 查找过程中所返回的一个域名，匹配源域，同时落在 ipamworldwide. com 域内，以及如果在 PTR 查找过程中所返回域名对应的 A/AAAA 域名，匹配电子邮件的源 IP 地址。

4）ip4＝验证源 IP 地址匹配所指定的 IPv4 地址；这种机制可由 CIDR 长度标出，如下例所示。

① ？ip4：192. 0. 2. 32/30。中性的，如果消息的源 IP 地址落在 192. 0. 2. 32～192. 0. 2. 35 内。

5）ip6＝验证源 IP 地址匹配所指定的 IPv6 地址；这种机制可由前缀长度标出，如下例所示。

① ＋ip6：2001：db8：f02b：2a：：/64。通过，如果消息的源 IP 地址落在 2001：DB8：F02B：2A：：/64 网络内。

6）exists：domain_ name＝查找对应于 domain_ name 的 A 记录（不是 AAAA 记录）；如果提供任意答案（IP 地址），则这种机制就得以匹配（这种机制必须以要匹配的一个域名进行范围限制），如下例所示。

① exists：ipamworldwide. com：匹配，如果 ipamworldwide. com 域的一条 A 记录查找返回一个 IP 地址的话。

7）include：domain_ name＝为了利用其 SPF 策略（例如，利用来自多个 ISP 之一个域或来自其他域（你是从这里发送电子邮件的）的策略），递归地评估 domain_ name。

8）all = 匹配任何东西；如果前面没有机制匹配就会失败的话，则-all 通常用作最后参数。

（3）修饰符。为了提高附加信息，则可在 SPF 记录内指定修饰符。修饰符是名字-值对，已经定义了其中两个。

1）redirect = domain_ name：使 SPF 记录的“别名法”将一个常见的 SPF 处理记录（比如）施用到多个域。这为正在发生的变更管理提供了方便：在一条记录中改变处理，最小化出现错误的机会，就最大化了一致性。在下例中，对 ipamworldwide. com 域的 MX 记录检查也将施用到 hq 和 euro 子域。

hq. ipamww. com. IN SPF “v = spf1 redirect = _ spf. ipamworldwide. com”

euro. ipamww. com. IN SPF “v = spf1 redirect = _ spf. ipamworldwide. com”

_ spf. ipamworldwide. com. IN SPF “v = spf1 + mx：ipamworldwide. com - all”

重定向可如上例一样显式使用，或作为“最后一道防线”，例如，列为最后侧的机制。

2）exp = domain_ name：解释，它定义这样的域，当一种机制匹配失败时，必须为其完成一条 TXT 记录查找，以便将字符串表示为结果。

（4）宏。从技术角度来看，上述机制中的任何一种机制和修饰符的 domain_ name 都不需要一个显式定义的（硬编码的）域，但可使用宏定义一个域，以便基于正被评估的消息信封形成一个域名。即使正由处理一个 exp 修饰符得到的 TXT 记录，也可用于宏。使用百分号（%）识别宏。如下宏已被定义。

1）%{s} = 发送者的电子邮件地址。

2）%{l} = 发送者的电子邮件地址的本地部分。

3）%{o} = 发送者的电子邮件地址的域。

4）%{d} = 当前域，通常与发送者的域相同，但也可能通过 include 机制（例如）已被处理。

5）%{i} = 消息发送者的源 IP 地址。

6）%{p} = 通过对消息发送者的源 IP 地址的 PTR 查找，经过验证的域名。

7）%{v} = 文字字符串，如果源 IP 地址是一个 IPv4 地址，为“in-addr”，如果源 IP 地址是 IPv6，则是“ip6”。

8）%{h} = HELO/EHLO 身份的域部分。

9）%% = 就是%（百分号）。

10）%_ = 空格“ ”。

11）%- = 一个 URL 编码的空格，例如“%20”。

如下宏可用于 TXT 记录，可由一个 exp 机制索引使用，但也可能没有其他地方使用。

1）%{c} = SMTP 客户端 IP 地址。

2）%{r} = 实施 SPF 检查的主机的域名。

3）%{t} = 当前时间戳。

宏变换器可利用一个宏的结果的一个子集（例如，通过确定域名标签的一个整

型量）或一个宏结果的反向结果（例如将一个 IP 地址反向）。反向做法即将一个 r 加入到宏的大括号（{}）之中。

（5）宏的范例。考虑图 10-3 的例子，其中迈克（mike@ ipamww. com）向我的 tim@ ipamworldwide. com（在 IP 地址为 192. 0. 2. 32 的 SMTP 主机上）发送一封电子邮件。使用来自该图的这个信息和其他信息，我们可将这个电子邮件传输的宏值定义为。

1）%{s} = mike@ ipamww. com。

2）%{l} = mike。

3）%{o} = ipamww. com。

4）%{d} = ipamww. com。

5）%{d3} = ipamww. com。

6）%{d2} = ipamww. com。

7）%{id1} = com。

8）%{i} = 192. 0. 2. 32。

9）%{ir} = 32. 2. 0. 192。

10）%{v} = in-addr。

11）%{h} = ipamww. com。

12）%{ir}. % {v} . _ spf. % {d} = 32. 2. 0. 192. in-addr. _ spf. ipamww. com。

SPF 为您的组织机构以不同粒度表示电子邮件策略，提供了一种强大的宏语言。但是，它是一个试验性的协议，可以将它看做 Sender ID 的一个近亲堂兄弟。

10.3.4 发送者 ID

识别可能是垃圾电子邮件的另一种试验性机制，被称作发送者 ID（Sender ID）。发送者 ID 机制寻求来自给定源 IP 地址处一个给定 SMTP 的一封给定电子邮件，是否是得到授权发送该电子邮件的。像 SPF 一样，发送者 ID 可依据 MAIL FROM 字段，检验电子邮件消息的发送者、发送者域和源 IP 地址。与 SPF 不同的是，发送者 ID 依据消息首部信息，也验证（或以替代方式）发送者和发送者域。发送者 ID 与 SPF 相似的是，它利用了 SPF 资源记录类型，如前一节所定义的情形，但作了一些修改。

1）版本字符串（“v = spf1”）替换为“spf2. 0”。

2）发送者 ID 包括记录的一个范围：“mfrom”和在 SPF 一样，指明 mailfrom 实体，并/或“pra”表明“所指的负责地址”（下面讨论）。

3）从 SPF 定义中扩展得到修饰符，这支持位置上下文，作为 SPF 定义的全局上下文的一种替代上下文。即，一个修饰符可影响前面的机制，这不像 SPF，在 SPF 中一个修饰符总是可全局施用的。

范围字段被用来推知发送者和发送者域，以便进行验证（即 MAIL FROM 实体和/或 PRA）。所指的负责地址 PRA，是与最接近于接收者电子邮件系统的发送者身份有关的范围。PRA 算法检查消息首部（而不是信封），并查找一个发送者地址，方法是按顺序检查如下首部，取所发现的第一个地址。

1）Resent-Sender（重发-发送者）首部。

2）Resent-From（重发-来源）首部。

3）Sender（发送者）首部。

4）From（来源）首部。

在这些首部之中发现的单一有效发送者邮箱地址（即 mailbox@ maildomain 的形式）是 PRA。在如图 10-3 和 10-4 所示的简单情形中，所指的负责地址将是 mike@ipamww. com，是从“From”首部值中推算得到的。一个第三方被用来代表一个合法的发送者来传输电子邮件，在这种情形中，将使用“Resent-From”或其他首部值。使用术语“所指的”，原因是该算法依赖于消息首部中提供的信息，而消息首部是由发送者提供的。

围绕发送者 ID 和 SPF 的使用，存在些微争论，这就是这两种技术都注定为试验性的方法的原因。例如，发送者 ID 对“v = spf1”（SPF）记录的处理可得到注定为垃圾邮件的有效消息。我们希望，在未来可得到一种一致的统一方法。

10.3.5 域名密钥可识别的邮件

域名密钥可识别的邮件（DKIM）指定了为电子邮件发送者以密码学方式对一封电子邮件消息进行签名的一种方法，以便使接收者通过发送者的域密钥的检索和应用，在接收到邮件消息时可对它进行验证。DKIM 利用数字签名（使一个给定的数据集合（在这种情形中是一条电子邮件消息）能够对数据签名，从而使那些接收到的数据和签名，使用一个对应的公开密钥，可对签名实施解密），可实施数据源发地和完整性验证。DKIM 采用一种非对称密钥对（私有密钥/公开密钥）模型。在这样一种模型中，电子邮件消息和选中的首部字段采用一个私有密钥进行加密，并可使用对应的公开密钥对数据解密，来实施验证。私有密钥和公开密钥形成一个密钥对。数学方面的细节是非常复杂的，但从概念上来说，私有/公开密钥对，为公开密钥的持有者提供了验证数据是使用对应私有密钥签名的一种方法。这就提供了所验证数据确实是由私有密钥的持有者签名的认证过程。数字签名也支持所接收数据匹配发布的数据且在中转过程中没有被篡改的验证。

参见图 10-5，数据源发者，如图左侧所示，产生一个私有密钥/公开密钥对，并利用私有密钥对数据签名。对数据签名的第一步是产生该数据的一个散列值，有时也称其为一个摘要。散列函数是单向函数[⊖]，为了更简单的操作，将数据加扰为一个固定长度的字符串，并代表数据的一个“指纹”。这意味着极不可能会出现另一个数据输入可得到相同散列值的情况。因此，经常将散列值用作校验和，但并不提供任何源发认证能力（知道散列算法的任何人均可简单地对任意数据进行散列运算）。常见的散列算法包括 HMAC-MD5、RSA-SHA-A 和 RSA-SHA-256。DKIM 不仅默认情况下使用 RSA-SHA-256，而且支持 RSA-SHA-1。使用私有密钥对散列值进

⊖ 一个单向函数指从散列值是不能唯一地推算得到原始数据的。人们可使用一个算法得到散列值，但不存在这样的反向算法，可对散列值实施运算得到原始数据。

行加密，以便得到签名。加密算法的输入是散列值和私有密钥，可产生签名。

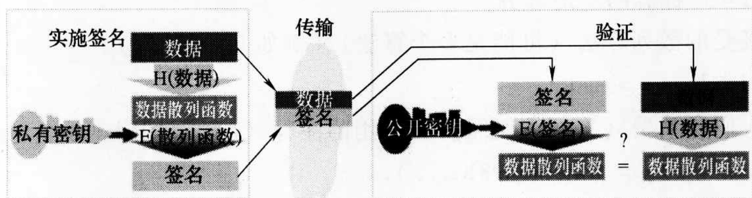


图 10-5　数字签名产生和验证的过程

消息及其关联的签名被传输到接收者。定义了一个新的电子邮件首部字段 dkim-signature，用来存储 DKIM 签名，还有检索公开密钥的信息也存储在该字段内。依据前边我们对 SMTP 工作原理的回顾，您可能会想，信封数据的修改和首部的插入会如何影响签名。DKIM 提供了规范化的一种“简单的”或严格的形式和一种“宽松的”形式。简单形式几乎不能容忍修改，而宽松形式可允许空格替换和首部行回绕，而不影响签名的有效性。

(1) DKIM 签名电子邮件首部字段。接收者必须从 dkim-signature 首部字段中抽取签名。dkim-signature 字段也包含如下信息。

1) DKIM 版本（例如 v = 1）。

2) 用来产生签名的算法（例如 a = rsa-sha256）。

3) 签名（例如 b = dqdVx0fAK9...）。

4) 规范化消息体的散列值（bh = 7Dkw0eE35Jlkjexcmpol...）。

5) 规范化方法（c = relaxed（宽松的））。

6) 签名的域标识符——签名实体的域（例如 d = ipamworldwide. com）。

7) 用户或代表用户的代理，对消息实施签名（i = rooney@ ipamworldwide. com）。

8) 在域内的选择器或密钥索引（允许每个域可有多个密钥，这在密钥轮换和更细粒度的签名中有所帮助）（例如 s = europe）。

9) 被签名首部字段的枚举（例如 h = from：to：subject：date）。

10) 其他可选的信息，包括用于检索公开密钥的查询方法。默认（当前唯一的）查询方法 q = dns/txt，指令接收者实施查询类型“txt”的一条 DNS 查询，检索对应于用来签名消息的私有密钥的公开密钥。令人关注的另一个可选字段 i = tag（标签）提供了用户的身份或代表用户签名消息的代理的身份。

(2) DKIM TXT 记录。使用查询方法 q = dns/txt，接收者在签名域内实施一条 TXT 记录的一次 DNS 查询。查询的问题节是这样形成的，将选择器值（s = 值）、字符串“_ domainkey”和指定的签名域（d = 值）串接。使用例子中一封进入的电子邮件 dkim-signature 字段中所指定的 s = europe 和 d = ipamworldwide. com，将发出查找 europe. _ domainkey. ipamworldwide. com 的一条 TXT 查询。对应 TXT 记录的 RData 部分包括类似于 dkim-signature 字段的一个或多个标签。

1) DKIM 版本（v = DKIM1）。

2）密钥的粒度，如果是指定的，则必须匹配 dkim-signature 首部（g = *）中用户或代理（i =）标志的本地部分。

3）所接受的散列算法（可能是多个算法）（例如 h = sha256）。

4）密钥类型（k = rsa）。

5）为人们所消费（方便人们阅读）使用的注释（n = updated_ key）。

6）公开密钥（p = Dkjeijf8d98Kz...）。

7）服务类型（s = email）。

8）标志，指明诸如 dkim-signature 首部中 i = 标签间的符合性规则、d = 域标签（在 TXT 记录中编码为 t = s）以及这个域是否正在测试 DKIM（t = y）。

唯一必备的标签是 p 标签，即公开密钥。一个范例 TXT 记录后根

europe. _ domainkey. ipamworldwide. com IN TXT

（"v = DKIM1；p = Dkjeijf98Kz..."）

在检索公开密钥时，接收者和源发信者一样，要计算所接收消息体的一个散列值，并对首部各字段签名。接收者使用源发者的公开密钥，对所接收到的签名使用散列算法。这种解密的输出是源发数据散列值，将其与接收者对数据所计算得到的散列值进行比较。如果这两者匹配，则数据没有被修改，是私有密钥持有者对该数据签的名。

如果进入电子邮件消息包含一个 dkim-signature 首部字段，则显然的是，发送者正在使用 DKIM，并对该消息签过名。但如果一封进入的电子邮件消息没有包含一个 dkim-signature 首部字段，这意味着发送者没有对消息签名吗？这事实上可能为一名垃圾邮件（SPAM）攻击者从一个伪造的源域发起未签名电子邮件消息打开了一个缺口。DKIM 依赖于作者域签名实践（ADSP）的发布，该 ADSP 使一台接收者电子邮件服务器能够确定是否应该依据策略对来自一个给定域的消息进行签名，且如果要签名的话，由谁签名以及带有哪些签名（可能有多个签名）。

一个接收者确定发送域的签名实践，方法是发出查找 Qtype = TXT 和 Qname = _adsp. _ domainkey. signing-domain-identifier 的一条查询，其中 signing-domain-identifie 同样是 d = 值。对应的 TXT 记录指明，来自这个域的电子邮件是总是被签名的、可能被签名的，总被签名的和任何未签名的电子邮件应该被丢弃。欲了解更多细节，请参见 RFC 5617[113]。

10.3.6 历史上出现过的电子邮件资源记录类型

在 DNS 的早期日子里，定义过这些资源记录类型，目前不再使用。我们将它们列在这里，纯粹出于历史意义的角度。

（1）MR——邮件重命名记录。MR 资源记录类型将电子邮件转换为一个个体（或多个人，每个人一条 MR 记录）的一个别名或列表。从最简单的含义来说，它提供了一个邮箱名的一个别名。

属主	TTL	类	类型	RData
电子邮箱别名	TTL	IN	MR	电子邮箱名
cfo	86400	IN	MR	finance(财务)

（2）MB——邮箱记录。MB 记录是在 RFC 1035 中定义的，并支持将一个用户 ID 与包含该用户电子邮箱的期望主机关联。

属主	TTL	类	类型	RData
电子邮件 ID	TTL	IN	MB	邮箱主机名
joe	86400	IN	MB	smtp. ipamworldwide. com

（3）MG——邮件组成员记录。RFC 1035 定义了 MG 资源记录，它支持将电子邮件用户与一个用户组关联。

属主	TTL	类	类型	RData
电子邮件组名	TTL	IN	MG	电子邮件 ID
finance(财务)	86400	IN	MG	joe

（4）MINFO——邮箱/邮件列表信息。MINFO 记录也定义在 RFC 1035 中，意图是提供邮箱和邮件列表信息。它提供两个电子邮箱地址，一个用于请求加入邮件列表，另一个用于报告错误。

属主	TTL	类	类型	RData
邮箱名	TTL	IN	MINFO	请求邮箱　错误邮箱
newsalerts	86400	IN	MINFO	hostmaster　majordomo

10.4　安全应用

10.4.1　保障名字解析的安全——DNSSEC 资源记录类型

第 13 章专门用来讨论 DNS 安全扩展（DNSSEC）话题，所以在本章内出于完备性考虑，将简单汇总一下 DNSSEC 必备的资源记录类型。在第 13 章我们将提供 DNSSEC 的全部上下文和描述。

（1）DNSKEY——DNS 密钥记录。DNSKEY 资源记录用于 DNSSEC，用来发布在区域信息上验证签名所用的公开密钥。服务器使用一个私有密钥，在一个区域内签名它的权威资源记录集合，对应的公开密钥则以 DNSKEY 记录形式在区域文件中发布。发布两种类型的密钥：一个区域签名密钥（ZSK）（它签名资源记录数据）和一个密钥签名密钥（KSK）（对 ZSK 签名）。解析器可使用这个公开密钥来验证一个给定的 RRSet 的签名。

属主	TTL	类	类型	RData
密钥名	TTL	IN	DNSKEY	标志　协议　算法　密钥
ipamww. com.	86400	IN	DNSKEY	256　3　5　AweE8F(le…

在这个例子中，RData字段解释如下。

1）标志字段提供有关密钥的类型和状态。当前为标志字段定义的值如下。

① bit7。这个密钥是一个区域签名密钥（十进制=256）。

② bit8。撤销这个密钥。

③ bit15。这个密钥是一个密钥签名密钥（十进制=1）。

④ 其他bit。未指派。

2）该协议字段必须有值“3”，指明是DNSSEC（这是当前定义的唯一一个值）。

3）在上例中算法字段有一个值“5”，指明RSA-SHA-1算法。当前支持的算法编码如下。

① 值=1。RSA/MD5，依据RFC 4034，不建议使用。

② 值=2。Diffie-Hellman。

③ 值=3。DSA㊀/SHA-1。

④ 值=4。为椭圆曲线保留。

⑤ 值=5。RSA-SHA-1，依据RFC 4034，这是必须支持的。

⑥ 值=6。DSA-NSEC3-SHA-1——算法3的一个别名，但带有NSEC3记录（而不是NSEC记录）使用的标识符。

⑦ 值=7。RSA-SHA-1-NSEC3-SHA-1——算法5的一个别名，但带有NSEC3记录（而不是NSEC记录）使用的标识符。

⑧ 值=8。RSA-SHA-256。

⑨ 值=10。RSA-SHA-256。

⑩ 值=12。GOST R 34.10-2001。

⑪ 值=252。间接的。

⑫ 值=253-254。私有的。

⑬ 值=0、123-251、255。保留的。

⑭ 其他值。未指派的。

4）密钥字段是公开密钥。

（2）DS——委派的签名人记录。RFC 4034[114]定义了DS资源记录类型，它本质上将信任链扩展到一个签过名的委托域（区域）。DS资源记录使一个父区域可认证它的子区域的公开密钥签名密钥（KSK的DNSKEY记录）。如此，则每个DS记录指向委托子区域中的一个特定（由密钥标签指明）DNSKEY资源记录。认证DS记录使各客户端可认证子区域的DNSKEY。

㊀ DSA=美国政府数字签名算法

属主	TTL	类	类型	RData
委托域	TTL	IN	DS	密钥标签 算法 类型 摘要
child. ipamww. com.	86400	IN	DS	32284 5 1 75CF28D3OQ35...

算法字段识别在对应DNSKEY记录上的算法字段。通过在摘要记录中包括DNSKEY RR的一个摘要（散列函数），DS记录指向一个DNSKEY记录；[摘要]类型字段指明用来构造摘要的算法。

(3) DLV——DNSSEC旁查（lookaside）验证记录。在RFC 4431[115]中规范定义，DLV资源记录用在DNSSEC内，用于在正常的DNS域树层次结构（即信任链）之外发布信任锚点。DLV记录是结构上等同于DS记录的，这是指它识别一个“代理的父区域”，且由此认证该“子”区域的公开密钥签名密钥记录（DNSKEY）。旁查验证的意图是在没有根和TLD区域签名的情况下，提供一种替代的上行信任锚点，例如dlv. isc. org。

属主	TTL	类	类型	RData
DLV域	TTL	IN	DLV	密钥标签 算法 类型 摘要
ipamww. com. dlv_reg. net	86400	IN	DLV	32284 5 1 90df80DF89lLe...

(4) NSEC——下一个安全记录。NSEC资源记录类型提供两个信息集合。在一个区域中的NSEC RR集合形成该区域中权威属主名的一个链，并指明在该区域中存在哪些权威的RRSet。NSEC资源记录包含下一个属主名（识别在该链内的关联权威属主名）和NSEC资源记录的属主名下存在的RR类型集合。

属主	TTL	类	类型	RData
RRSet属主	TTL	IN	NSEC	下一个RRSet属主 类型比特映射
ns1. ipamww. com.	86400	IN	NSEC	ns2. ipamww. com. A NS RRSIG NSEC

下一个RRSet属主字段包含以区域规范顺序排列的下一个属主名（它有权威数据），或包含定义一个委托点的一个类型NS的RRSet。这在NSEC属主字段内已识别RRSet和下一RRSet属主RData字段之间资源记录存在性的已认证否定结果。类型比特映射字段识别存在于这个NSEC资源记录属主名下的资源记录类型。在这个字段内，如果一个bit=1，那么对应于这个bit号的RRType就是存在的。因此如果bit1为1，则对应于RRType=1或A记录，那么就存在一个A RRSet。幸运的是，这个内容的文本表示位于人们熟悉的资源记录类型助记符范围内。

(5) NSEC3——NSEC3记录。NSEC资源记录提供了对RRSet存在性的经过认证的拒绝答复，但它也支持轻易将区域中RRSet枚举出来，这被认为是一个信息安全风险。换句话说，一名好奇的或恶意的查询者可尝试解析一个伪造的名字，并接收到包含被查询主机名的资源记录属主名列表。

像NSEC一样，NSEC3记录提供经过认证的RRSet存在性拒绝答复，但使区域中的RRSet链有点混乱。这种混乱使一个区域的内容的指纹操作，从计算角度看更加密

集。NSEC3 并不指向新的属主名字段，相反，它指向散列序中的下一个经过散列处理的属主名字段。在散列值产生之前添加到每个属主名的精选值（salt value）使某人尝试对区域进行指纹操作，产生经过散列的属主名字更加复杂。

对于区域中的每个 RRSet，使用可施用于属主名的指定散列算法，对属主字段进行散列处理，并与 <Iterations（重复次数）> +1 次的 salt（精选）字段串接在一起。如下伪码以另一种方式对此进行了说明：

```
x = {与精选值串接的 RRSet 属主字段}
y = H(x)   x 的一个散列值,依据前面语句定义的算法实施计算
for(i = 重复值; i > 0; i -) {
      y  =  H (y)
}
```

属主	TTL	类	类型	RData
经过散列处理的 RRSet 属主	TTL	IN	NSEC3	散列算法　标志　重复次数　精选长度　精选值　散列长度 下一个经过散列处理的属主名　类型比特映射
jAdfJE; ...	86400	IN	NSEC3	0 2 8 a808f6ce1a950b1c 18 k0Lse7… A RRSIG NSEC3

NSEC3 记录的 RData 字段定义如下：

1）散列算法。用来构造散列值的算法；有效值是

① 保留的。

② 1 = RSA-SHA-1。

③ 2 ~255。未指派的。

2）标志。由一组 8 个布尔标志组成，标志字段目前有一个已定义的单一标志 (bit0)。如果 bit0 被设置，则这表明这个记录涵盖了一个或多个未签名的委托记录。这个决定退出标志使保障委托安全“退出”到未签名的区域（即验证一个子区域的 DS 记录是不存在的）。

3）重复。指定散列函数的附加施用次数。

4）精选长度。包括在传输（wire）格式内，但不存在于资源记录文本格式内，这个字段指明以字节表示的精选字段长度（有效值 =0 ~255）。

5）精选（Salt）值。在应用散列函数之前，将精选字段的值附加在 RRSet 之后，并以不区分大小写的十六进制方式表示。

6）散列长度。以字节表示的下一个经散列处理过的属主名字段的长度，包括传输的格式，但不以资源记录文本格式表示。

7）下一个经散列处理过的属主名。

8）类型比特映射。这个字段在区域内为这个属主定义了资源记录类型，并以 NSEC 记录中相应字段的相同形式进行编码。

（6）NSEC3PARAM——NSEC3 参数记录。NSEC3PARAM 记录类型定义了计算散

列属主名所需的参数以及区域内要签名的对应 NSEC3 记录。服务器使用 NSEC3PARAM 记录，来识别对一条查询做出响应的否定答案。由此，当一条查询到达，要查询区域内一条不存在的 RRSet 时，服务器将施用 NSEC3PARAM 参数，对被查询的属主名进行散列处理，目的是提供一条合适的 NSEC3 响应，即这条被查询的属主名落在哪两个散列处理过的 RRSet 之间？在区域文件内应该仅存在一条 NSEC3PARAM 记录。当服务器自动地对新的 RRSet 或改变的 RRSet 签名时，服务器也使用 NSEC3PARAM 记录。

RData 字段和 NSEC3 RData 字段内的相应字段一样具有相同的含义。

属主	TTL	类	类型	RData
域名	TTL	IN	NSEC3PARAM	散列算法 标志 重复次数 精选长度 精选值
ipamww. com.	86400	IN	NSEC3PARAM	1 0 2 8 a808f6ce1a950b1c

（7）RRSIG——资源记录集签名记录。资源记录集签名资源记录包含与一个给定 RRSet 关联的数字签名。这个签名，与区域的公开［区域签名］密钥一起，用来认证相应的 RRSet 的完整性和来源。

属主	TTL	类	类型	RData
RRSet 属主	TTL	IN	RRSIG	涵盖类型 算法标签 原始 TTL 签名超时 签名开始时间 密钥标签 签名者 签名
ftp1. ipamww. com.	86400	IN	RRSIG	A 5 3 86400 20080515133509 20080515133509 27783 ipamww. com. N78E…

RRSIG 记录内部的 RData 字段定义如下：

1）涵盖类型。由这个签名所签署的相应属主和类的资源记录类型。这个字段是本章通篇针对资源记录所讨论的标准资源记录类型。在上例中，A（地址）资源记录类型指明，类 IN 的带有名字 = ftp1. ipamww. com（属主字段）的一条记录是使用这条 RRSIG 记录进行签名的。

2）算法。出于与所接收签名的比较目的，这个算法用于产生数据的散列值。这个字段以 DNSKEY 资源记录类型的算法字段相同的方式进行编码。

3）标签。指明标签数量。回顾一下，标签是指向域名的文本表示的，每个名字“位于点号之间”都有一个标签。因此，www. ipamworldwide. com 有三个标签。这个字段用来重构原始属主名，用来在如下情形中产生签名，即由服务器返回的属主名有一个通配标签（*）。

4）原始 TTL。在权威区域中定义的签名 RRSet 的 TTL，用来验证一个签名。需要这个字段，原因是在原始响应中返回的 TTL 字段，正常情况下要由一个缓存解析器做减一处理，使用那个 TTL 值可能导致错误的计算。

5）签名超时。这个签名超时时的日期和时间，表示为自 1970 年 1 月 1 日 00：00：00 UTC 以来的秒数，或表示为 YYYYMMDDHHmmSS 的形式，其中。

① YYYY 是年。

② MM 是月，01 ~ 12。

③ DD 是月中的日期，01 ~ 31。

④ HH 是 24h 表示法中的 h，00 ~ 23。

⑤ mm 是 min，00 ~ 59。

⑥ SS 是 s，00 ~ 59。

⑦ 在这个日期/时间之后，签名是无效的。

6）签名开始时间。这个签名开始时的日期和时间，其格式化的方法与签名超时字段的方式相同。在这个日期/时间之前，签名是无效的。

7）密钥标签。提供与相应 DNSKEY 资源记录的一个关联，该资源记录可被用来验证签名。

8）签名者的名字。识别 DNSKEY 资源记录的属主名（即域名），该资源记录可被用来验证这个签名。

9）签名。涵盖该资源记录集的密码学签名，资源记录集合由这个 RRSIG 属主、类所定义，包括类型字段和这 RRSIG RData 字段（排除这个签名字段）。

10.4.2　其他面向安全的 DNS 资源记录类型

（1）TA——信任权威记录。虽然不存在定义 TA 资源记录的一个 RFC，但 IANA 已经为它分配了一个值，所以我们在这里描述一下。TA 资源记录在格式上与 DS 记录类型的格式相同，包括密钥标签、算法、摘要类型和摘要的 RData 字段。使用 TA 记录的做法，使一个解析器能够有一个资源记录签名，该签名由一个已知的信任权威验证，即使根区域没有被签名的情况下也是如此（它现在已被签名）。现在使用 DLV 记录提供这个功能。

（2）CERT——证书记录。RFC 4398[116] 定义了 CERT 记录，作为 DNS 中存储证书和证书撤销列表（CRL）的一种方法。证书提供了识别一个组织机构、服务器、个人或其他实体的一种方式，并将一个公开密钥与那个身份相关联。公开密钥可被用来认证发送者的身份，对通信进行加密和解密，并验证消息的完整性。证书是层次结构的，并可被用来到一个已知信任实体（证书权威）的（关系）进行验证。CRL 是证书的列表，由于超时或人工撤销，这些证书已被撤销。

包含证书的 CERT 记录被存储在 DNS 之中，使解析器可通过 DNS 得到证书，而不是从一个目的地证书服务器处得到证书。CERT 资源记录有如下格式。

属主	TTL	类	类型	RData
域名	TTL	IN	CERT	证书类型　密钥标签　算法　证书或 CRL
ipamww. com.	86400	IN	CERT	PGP 436 3 A4df480DFC9lLa...

当在该记录的 RData 部分包括一个证书时，属主字段识别证书所施用的实体。如果一个 CRL 被包括在 RData 节中，则属主名应该包含与发行权威有关的域名。RData 部分包含如下子字段。

1）证书类型。例如 X. 509/PKIX、PGP 以及其他。

2）密钥标签。用来加速到那些匹配密钥标签的相关证书的识别过程。

3）算法。在阐述密钥中使用的算法，它以 DNSKEY 资源记录类型中算法字段相同的方式进行编码。

4）证书或 CRL。

（3）IPSECKEY——IPSec 记录的公开密钥。在 RFC 4025[117] 中定义的 IPSECKEY 资源记录类型，提供了一种在 DNS 中存储一个公开密钥的方式，该密钥与 IPSEC 一起使用。这个资源记录一种使一个客户端能够寻求与一台远端主机建立一条 IPSec 隧道，用来识别认证该远端主机的方式，并确定是与该主机是直接连接，或通过作为一个网关的另一个节点进行连接。IPSECKEY 资源记录与预期远端主机的 IP 地址或主机域名关联。IP 地址被存储在 . arpa. reverse 域空间之中。IPSECKEY 资源记录的格式如下。

属主	TTL	类	类型	RData
在 . arpa. 域中的 IP 地址或主机域名	TTL	IN	IPSECKEY	优先级　网关类型　算法　网关　公开密钥
1. 0. 12. 10. in-addr. arpa.	86400	IN	IPSECKEY	10 1 2 10. 100. 1. 2 Adf4C91L. . .

RData 字段包含如下字段。

1）优先级。用来对一个常见 RRSet 内的多条记录确定优先级，使用最低优先级优先的方法。

2）网关类型。指明网关字段的格式。

① 0 = 不存在网关。

② 1 = IPv4 地址。

③ 2 = IPv6 地址。

④ 3 = FQDN。

3）算法。公开密钥字段的格式。

① 0 = 不存在密钥。

② 1 = DSA 格式的密钥。

③ 2 = RSA 格式的密钥。

4）网关。标识一个网关，为了到达远端主机（由属主字段标识）而与该网关建立一条 IPSec 隧道。这个字段的解释受到网关类型字段的控制。

5）公开密钥。所产生的密钥，使用算法字段中指定的算法产生。

（4）KEY——密钥记录。密钥记录使用 DNSSEC 的初始典型（incarnation）加以定义，但由 DNSKEY 资源记录取代。但是，在 DNSSECbis 发布之前，密钥记录也被用来存储与 SIG（0）记录关联的公开密钥。密钥记录与 DNSKEY 记录具有相同的格式。

属主	TTL	类	类型	RData
密钥名	TTL	IN	KEY	标志 协议 算法 密钥
K3941. ipamww. com.	86400	IN	KEY	256 3 1 12S9X-weE8F(le...

（5）KX——密钥交换器记录。KX 记录支持一个中介的指定，它可代表另一台主机提供一个密钥。换句话说，如果打算与 x. ipamworldwide. com 进行密钥协商，则 KX 记录可指向 y. ipamworldwide. com 主机域名，应该与其进行密钥交换协商。一个优先级字段支持多个替代域的指定，它们具有进行密钥协商的不同优先级。

属主	TTL	类	类型	RData
主机域名	TTL	IN	KX	优先级 密钥交换器主机域名
x. ipamworldwide. com.	86400	IN	KX	10 y. ipamworldwide. com.
x. ipamworldwide. com.	86400	IN	KX	20 z. ipamworldwide. com.

（6）SIG——签名记录。在 DNSSEC 范围内，由 RRSIG 记录取代 SIG 资源记录，但在 DNSSEC 范围之外，SIG 记录仍然用于数字化地签名 DNS 更新和区域传递。即，您不需要部署 DNSSEC 来支持更新和区域传递的事务签名。这种事务可通过 TSIG（事务签名）记录使用共享的秘密密钥进行签名，或通过 SIG（0）使用私有/公开密钥进行签名，在后一种方法中对应的公开密钥被存储为 KEY（密钥）记录。SIG（0）表示法指使用带有一个空（0）类型涵盖字段的 SIG 资源记录。在这样的情形下，RFC 2931（118）建议将属主字段设置为根、TTL 设置为 0、类设置为 ANY，如下例所示。

SIG 记录的格式等同于 RRSIG 记录的格式，例外是超时日期和开始日期字段的格式有所不同；对于 SIG 记录，这些字段不是依据 RRSIG 记录的日期格式的，相反将其格式定为一个递增的整数，列举自 1970 年 1 月 1 日 00：00：00 UTC 以来的秒数。这个计数器将会返回到 0，并在计数器超过 42.9 亿秒（136 年多一点）之后，继续计数。

属主	TTL	类	类型	RData
RRSet 域	TTL	IN	SIG	涵盖类型 算法 标签 原始 TTL 签名超时 签名开始 密钥标签 签名者 签名
.	0	ANY	SIG	0 3 3 86400 20080515133509 20080515133509 26421 ipam-ww. com. Zx9v...

（7）SSHFP——安全的 Shell（外壳）指纹记录。安全的外壳（SSH）协议支持在一个不安全的 IP 网络上，从一台客户端到一台服务器和其他安全网络服务的安全登录。链接的安全性依赖于用户向服务器认证他或她自己，以及服务器向客户端认证服务器自己，使用的是 Diffie-Hellman 密钥交换。如果客户端还不知道公开密钥，则为了验证用户，服务器提供密钥的一个指纹。在 DNS 中存储这个密钥指纹，为客户端通过一个“第三方”以带外方式查找并验证指纹，提供了一种方法。查找要求使

用 DNSSEC，来保障查找过程的安全，并确保消息完整性。SSHFP 资源记录是用来存储这些 SSH 指纹的记录类型。

属主	TTL	类	类型	RData
主机域名	TTL	IN	SSHFP	算法 指纹类型　指纹
srv21. ipamww. com.	86400	IN	SSHFP	2 1 8Fd7q90Dtfd...

SSHFP 记录的 RData 部分包括如下字段。

1）算法。当前定义的值如下。

① 0 = 保留的。

② 1 = RSA。

③ 2 = DSA。

2）指纹类型。当前定义的值如下。

① 0 = 保留的。

② 1 = SHA-1。

3）密钥指纹。

10. 4. 3　地理定位查找

（1）GPOS——地理位置记录。最初在 RFC 1712[119] 中定义的 GPOS 资源记录类型，已经被 LOC 资源记录类型所替代。GPOS 对一台主机的经度、纬度和高度编码，如下所示。

属主	TTL	类	类型	RData
主机域名	TTL	IN	GPOS	经度 纬度 高度
srv1. ipamww. com.	86400	IN	GPOS	39. 582 -75. 801 128. 2

（2）LOC——定位资源记录。这种类型的资源记录支持对相应主机的经度、纬度和高度信息进行编码。RFC 1876[120] 定义了 LOC 记录，这就废弃了 GPOS 资源记录类型。LOC 记录的 RData 字段给出了三个维度中的每个坐标。

1）纬度。°（“′”“″”）“N”或“S”。

2）经度。°（“′”“″”）“E”或“W”。

3）高度。以 m 为单位的高度。

4）每个测度的准确度，以 m 为单位的“误差球面”的直径。

属主	TTL	类	类型	RData
主机域名	TTL	IN	LOC	纬度　经度　高度　准确度
srv-97. ipamww. com.	86400	IN	LOC	39 58 N　75 38 W　128　50m

在上面的例子中，主机名为 srv-97. ipawmww. com 的机器位于北纬 39°58′、西经 75°38′，海拔 128m 处，精确度都在直径为 50m 的误差球面内。

10.4.4 非IP主机地址查找

（1）ISDN——综合业务数字网记录（试验型的）。该ISDN类型支持将一个ISDN地址与一台主机关联。ISDN地址的形式为一个电话号码，由国际电信联盟标准E.164定义。子地址字段是可选的。

属主	TTL	类	类型	RData
主机域名	TTL	IN	ISDN	ISDN地址 子地址
isdnhost. ipamww. com.	86400	IN	ISDN	16105551298 318

（2）NSAP——网络业务接入点记录。NSAP资源记录支持将一个主机名或FQDN翻译为一个网络业务接入点（NSAP）地址。NSAP是一台网络设备的表示法，该设备支持ISO无连接网络协议（CLNP）。不需要了解NSAP地址的细节，这种地址从来就没有真正被人们所理解，NSAP资源记录的功能等价于IPv4的一个A记录和IPv6的AAAA记录。它为一个被查询的主机名提供一个目的地址。

属主	TTL	类	类型	RData
主机域名	TTL	IN	NSAP	NSAP地址
nsap-host. ipamww. com.	86400	IN	NSAP	47. 0005. 09. d78d01. 1010. 0ffe. 0011... 00

（3）NSAP-PTR——网络业务接入点反向记录。NSAP-PTR记录类型执行NSAP地址的等价指针记录功能，它将一个NSAP地址后缀链接到一个主机域名。nsap. int域作为相应反向TLD。就和基于IP地址的指针记录一样，NSAP地址必须被反向，在每个数字之间插入点号。最后，添加nsap. int. 后缀。

属主	TTL	类	类型	RData
被反向的NSAP地址	TTL	IN	NSAP-PTR	主机域名
0. 0... 1. 1. 0. 0. e. f. f. 0. 0. 1. 0. 1. 1. 0. d. 8. 7. d. 9. 0. 5. 0. 0. 0. 7. 4. nsap. int.	86400	IN	NSAP-PTR	Nsap-host. ipamww. com

（4）PX——X.400的指针。PX资源记录是在RFC 2163[121]中定义的，意图是在DNS域名和一个X.400地址之间提供一个映射，用于电子邮件地址映射。X.400是消息通信或电子邮件的一个OSI标准，但如今多数系统都使用简单邮件传递协议。这个资源记录类型用于包含SMTP到X.400电子邮件网关的网络，该网关也被称作MIXER（MIME因特网X.400增强型中继）网关。使用源发者/接收者（O/R）惯例形成X.400地址。

属主	TTL	类	类型	RData
域名	TTL	IN	PX	优先级 DNS域 X.400映射
ipamww. com.	86400	IN	PX	10 ipamww. com. O = company. PRMDnetx. ADMD. C = tv.

（5）X25——X.25 PSDN地址记录（试验型的）。这是一个试验型的资源记录且

没有被广泛使用，原因是 X.25 报文交换数据网络（PSDN）如今没有被广泛使用。它具有许多可能应用。

1）记录在 IP 到 X.25 和 SMTP 到 X.25 的静态配置中所用的地址。

2）自动地将一个 IP 地址与 PSDN 地址相关联。

3）将名字配置到 X.25 PSDN 地址。

它也为广域非广播网络提供了一项类似 ARP 的功能。

属主	TTL	类	类型	RData
主机域名	TTL	IN	X25	PSDN 地址
x25-host.ipamww.com	86400	IN	X25	31161700956

（6）RT——路由通过。路由通过资源记录是在 RFC 1183[108] 中定义的，并被用于指明一个代理或替代目的地，在没有一条直接网络链路条件下，将主机的流量路由到该代理或替代目的地。可识别确定多个路由通过主机，每个都带有关联的优先级值，这很像 MX 资源记录。

属主	TTL	类	类型	RData
主机域名	TTL	IN	RT	优先级 代理主机名
host.ipamww.com.	86400	IN	RT	10 proxy.ipamww.com.

10.4.5　Null 记录类型

NULL（空）资源记录类型是试验型的，并支持指定多达 65535B 的“任何东西”。通常它是可被忽略的，且不被广泛使用。

属主	TTL	类	类型	RData
主机域名	TTL	IN	NULL	多达 65535 个字节的“任何东西”
host.ipamww.com	86400	IN	NULL	“Ignore this NULL resource record!”（忽略这条 NULL 资源记录）

10.5　试验型的名字-地址查找记录

10.5.1　IPv6 地址链——A6 记录（试验型的）

考虑到 IPv6 地址的绝对（sheer）长度，IETF 考虑了将主机名解析到 IPv6 地址的一种迭代方法。在 RFC 2874[122] 中定义的 A6 记录，其意图是将一个主机域名映射到一个 IPv6 地址的一部分（或所有部分），带有各个指针，用于使解析器迭代地将 IPv6 地址的剩余部分解析到完整的 128bit。这就支持了主机域名的解析，方法是从最常见的接口 ID 开始，之后添加合适的子网 ID 和全局路由前缀，本质上是从右到左地移动，解析主机名地址。其意图是简化 IPv6 网络的重新编址，由于网络维护、ISP 变更

或其他原因，这种做法可能是必要的。为许多台主机变更子网 ID 就和变更一条记录一样简单，所以不用改变每台主机的记录。

但是，由于以合适的链接关系（并防止开放的链接）而准确地配置 DNS 的复杂性，这条记录类型被更改为试验状态。为了形象地说明这点，下面的例子形象地说明 A6 资源记录，并说明后续的三条查询如何被用来进行完全解析。注意，基于个体优先级，可定义更多的或更少的链接关系。

属主	TTL	类	类型	RData
主机域名	TTL	IN	A6	前缀长度 地址后缀 前缀名
ftp-sf. ipamww. com.	86400	IN	A6	64 ::A05F:0:0:2001 sf-net. ipamww. com.
sf-net. ipamww. com.	86400	IN	A6	64 0:0:0:8400:: na-west. ipamwwe. com.
na-west. ipamww. com.	86400	IN	A6	48 2001:DB8:4AF0::

注意 A6 资源记录的 RData 部分包含三个子字段。前缀长度指明从地址开始到开始插入地址后缀比特这段范围内偏移比特的数量。因此，带有属主字段“ftp-sf. ipamww. com.”的第一个列出的 A6 记录，指明一个 64bit 的前缀长度，指定了接口标识符为:: A05F：0：0：2001。

前缀名字段提供了到第二次查找的一个链接关系，目的是继续构造完整的 128bit 地址。在这种情形中，我们链接到“sf-net. ipamww. com.”前缀，它指向带有属主字段“sf-net. ipamww. com.”的一条 A6 记录。对应的 A6 记录指明带有 IPv6 地址 0：0：0：8400::的一个 48bit 前缀长度。注意这里使用了完备 IPv6 地址表示法，它包括对单一双冒号的约束。之后这条记录指向 na-west. ipamww. com. A6 记录，这就以零偏移针对解析完成了我们的 IPv6 地址构成。图 10-6 形象地说明了这个过程。

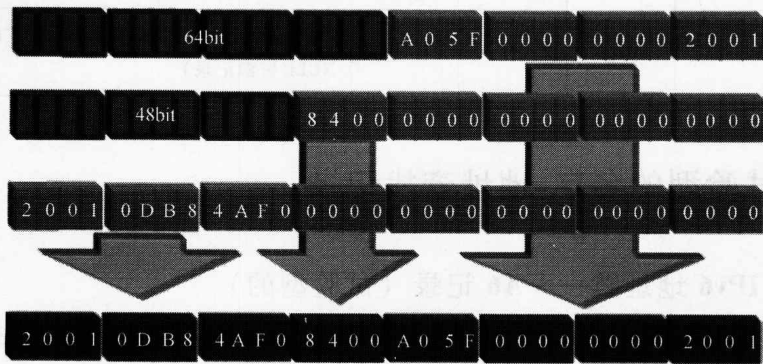


图 10-6　使用 A6 记录，对一个 IPv6 地址的迭代推算

10.5.2　APL——地址前缀列表记录（试验型的）

A 和 AAAA 记录是用来解析主机 IP 地址的，而 APL 记录寻求解析地址前缀或子

网地址。如下例子形象地说明了这样一个场景，它通告与一个域或主机关联的一个地址范围集合。APL 记录的 RData 部分由一个可选的感叹字符（!）、地址族（由 IANA[⊖]定义）后跟一个冒号，之后是 CIDR 表示（网络/前缀长度）组成。

属主	TTL	类	类型	RData
主机域名	TTL	IN	APL	地址族:地址/前缀
sf-ftp. ipamww. com.	86400	IN	APL	1:10. 0. 128/18，! 10. 16. 128. 0/18 2:2001:DB8:4AF0:8400::/56

10.6 资源记录小结

表 10-1 汇总了当前定义的资源记录集合，它依据资源记录类型的字母顺序排列（RRType——当一个查询器从 DNS 中寻找这种类型的信息时，它也对应于有效的 QType，即在一条 DNS 消息的问题节内）。虽然不是所有的资源记录都是 IETF 标准或甚至定义在 IETF 内，但这些资源记录的多数记录都由 IANA 指派了一个 RR 类型 ID 号，它们都在此列出。和所定义文档一起也给出了当前的 IETF 状态，可获取这些文档以便了解更多细节。

表 10-1 资源记录和查询类型小结

RRType(或 QType)	RR 目的(即 RData 内容)	RR 类型 ID	IETF 状态	定义的文档
A	一个给定主机名的 IPv4 地址	1	标准	RFC 1035[99]
AAAA	一个给定主机名的 IPv6 地址	28	草案标准	RFC 3596[123]
A6	一个给定主机名的 IPv6 地址或迭代 IPv6 地址解析的地址一部分	38	试验型的	RFC 2874[122]
AFSDB	一个给定 AFS 和 DCE 域的服务器主机名	18	试验型的	RFC 1183[108]
APL	一个给定域的地址前缀列表	42	试验型的	RFC 3123[124]
ATMA	一台主机的异步传递模式(ATM)地址	34	没有提交	由 ATM 论坛发布的 ATM 名字系统规范[125]
CERT	证书或证书撤销列表	37	标准跟踪	RFC 4398[116]

⊖ 地址族值是由 IANA 维护的，见 http：//www. iana. org/assignments/address-family-numbers。与我们的例子有关的是，IANA 向 IPv4 指派族号为 1、向 IPv6 指派族号为 2。

（续）

RRType（或 QType）	RR 目的（即 RData 内容）	RR 类型 ID	IETF 状态	定义的文档
CNAME	一台主机的主机名别名	5	标准	RFC 1035[96]
DHCID	将一个 DHCP 客户端的身份与一个 DNS 名关联	49	标准跟踪	RFC 4701[126]
DLV	一个信任锚点的权威区域签名	32769	信息型的（DNSSEC）	RFC 4431[115]
DNAME	域名别名	39	建议标准	RFC 2672[107]
DNSKEY	在一个信任链内的权威区域签名	48	标准跟踪（DNSSEC）	RFC 4034[114]
DS	被委托子区域的签名	43	标准跟踪（DNSSEC）	RFC 4034[114]
GID	组 ID	102	RESERVED（保留）	IANA-保留的
GPOS	一台给定主机的纬度/经度/高度——由 LOC 替代	27	试验型的	RFC 1712[119]
HINFO	一台主机的 CPU 和 OS 信息	13	标准	RFC 1035[99]
HIP	主机身份协议	55	试验型的	RFC 5205[127]
IPSECKEY	一个给定 DNS 名的公开密钥，用于 IPSec	45	建议标准	RFC 4025[117]
ISDN	一台给定主机的综合业务数字网（ISDN）地址和子地址	20	试验型的	RFC 1183[108]
KEY	在 DNSSEC 内由 DNS-KEY 替代，但仍为 SIG（0）和 TKEY 所用	25	建议标准	RFC 2536[128]
KX	为给定域中一台主机得到一个密钥的中间域	36	信息型的	RFC 2230[129]
LOC	一台给定主机的纬度/经度/高度和精度	29	不常见的	RFC 1876[120]
MB	一个给定电子邮件 ID 的邮箱名	7	试验型的	RFC 1035[99]
MD	一个给定域的邮件交付主机	3	过时的	RFC 1035[99]
MF	为了将邮件转发到一个给定域，将接收邮件的主机	4	过时的	RFC 1035[99]

（续）

RRType（或 QType）	RR 目的（即 RData 内容）	RR 类型 ID	IETF 状态	定义的文档
MG	一个给定电子邮件 ID 的邮件组邮箱名	8	试验型的	RFC 1035[99]
MINFO	为一个给定邮箱名发送账户请求或错误报告的邮箱名	14	试验型的	RFC 1035[99]
MR	一个邮箱名的别名	9	试验型的	RFC 1035[99]
MX	电子邮件主机解析的邮件交换器	15	标准	RFC 1035[99]
NAPTR	用于 DDDS、ENUM 等应用的一个通用字符串的统一资源标识符	35	标准跟踪	RFC 3761[111]
NS	一个给定域名的名字服务器	2	标准	RFC 1035[99]
NSAP	一台主机的网络服务接入点地址	22	不常见	RFC 1706[130]
NSAP-PTR	一个给定 NSAP 地址的主机名	23	不常见	RFC 1706[130]
NSEC	用于 DNSSEC 的一个资源记录集合的经过认证的确认或存在性否定确认	47	标准跟踪（DNSSEC）	RFC 4034[114]
NSEC3	用于 DNSSEC 的一个资源记录集合存在性的经过认证的否定确认	50	标准跟踪（DNSSEC）	RFC 5155[131]
NSEC3 PARAM	用于计算散列属主名的 NSEC3 参数	51	标准跟踪（DNSSEC）	RFC 5155[131]
NULL	一台给定主机的任何东西，多达 65535B	10	试验型的	RFC 1035[99]
NXT	由 NSEC 替换	30	过时的（DNSSEC）	RFC 3755[132]
PTR	一个给定 IPv4 或 IPv6 地址的主机名	12	标准	RFC 1035[99]
PX	一个给定域名的 X.400 映射	26	不常见	RFC 2163[121]
RP	一台主机的电子邮件地址和用于更多信息的 TXT 记录指针	17	试验型的	RFC 1183[108]

（续）

RRType(或 QType)	RR 目的(即 RData 内容)	RR 类型 ID	IETF 状态	定义的文档
RRSIG	一个给定域名、类和 RR 类型的资源记录集合的签名	46	标准跟踪(DNSSEC)	RFC 4034[114]
RT	一台给定主机的代理主机名,该主机并不总是处于连接状态的	21	试验型的	RFC 1183[108]
SIG	由 DNSSEC 内的 RRSIG 替换;由 SIG(0)和 TKEY 使用	24	建议标准	RFC 2536[128]
SOA	一个区域的权威信息	6	标准	RFC 1035[99]
SPF	发送者策略框架,使一个域属主能够识别这样的主机,它们被授权从该域发送电子邮件	99	试验型的	RFC 4408[112]、RFC 4409[133]
SRV	在一个域中提供指定服务的主机	33	标准跟踪	RFC 2782[134]
SSHFP	安全外壳指纹,支持使用 DNSSEC 的 SSH 主机密钥验证	44	标准跟踪	RFC 4255[135]
TXT	与一台主机相关联的任意文本	16	标准	RFC 1035[99]
UID	用户 ID	101	保留	IANA 保留
UINFO	用户信息	100	保留	IANA 保留
UNSPEC	未指定的	103	保留	IANA 保留
WKS	在一个指定 IP 地址处通过一个给定协议可以使用的服务,如今更普遍地用于一台主机的 SRV RR	11	标准	RFC 1035[99]
X25	X.25 PSDN	19	试验型的	RFC 1183[108]

第 11 章 DNS 服务器部署策略

本章[⊖]详细讨论 DNS 服务器的部署策略和折中考虑因素。一般而言，相比 DHCP，DNS 服务器支持更加面向角色的部署，我们的讨论将配置与特定角色（外部解析、缓存、内部使用等）相关联。当然，预算资金和更加靠近端用户的服务器数量快速增长之间的常见折中考虑，就像在 DHCP 一样，也同样适用于 DNS。

和 DHCP 部署一样，DNS 部署设计应该考虑到高可用性、性能和安全。为了分割名字空间和解析的职责界限，使用一种组件构造块方法进行 DNS 服务器部署，能够有助于取得这些通用目标，在本章通篇我们将讨论这样一种方法。要牢记的是，在定义一个“均码”型架构中，不存在曲奇饼成型刀式的 DNS 部署方法。但是，通过将基于角色的服务器配置定义为部署构造块，您就能够选择哪些是适用于您环境的规模和政策。

11.1 通用的部署指导原则

要牢记在心的一些通用原则包括如下。

1）对任何给定区域或区域集合，部署一台主权威服务器和至少两台从属权威服务器。

2）对于微软 Windows DNS 服务器部署方法，为每个域部署多个域控制器。

3）为了做到站点多样化的高可用性，在不同子网上（理想情况下，在不同位置）部署权威服务器。

4）为了得到较佳的性能和较低的网络额外负担，在“靠近”客户端/解析器处部署权威服务器。对于外部的服务器，在靠近因特网连接处部署；对于内部服务器，在较接近较高密度雇员区域部署。

5）为主服务器考虑部署一种冗余的硬件解决方案。

6）为处理外部查询和内部查询，应该部署独立的 DNS 服务器。就外部查询而言，我们指来自组织机构外部（例如因特网）的那些查询。内部查询是指来自组织机构内部的那些查询。

7）考虑独立的服务器，它们负责解析这样的权威数据，这些数据来自于负责解析递归查询的那些服务器（代表桩解析器）。

⊖ 本章的多数内容依据的是与 Alex Dresher[165] 的谈话和他的私人文档。

11.2 通用的部署构造块

本节以组成构造块的方式提供了常见 DNS 部署场景的一个概述。依据一个查询来源（查询源）和被查询信息的范围（查询范围），我们将这些构造块分解成四个大类。我们按照上述定义查询源，外部查询来自于公开因特网，而内部查询来源于组织机构内部。查询范围通常遵循这种通用分解方法，即外部范围处理因特网可达的解析数据，内部范围包括组织结构内的解析信息。下表小结了这种分类，使用的恰恰仅是原始分类名。

查询源	查询范围	
	外部	内部
外部	外部-外部	外部-内部
内部	内部-外部	内部-内部

另外，我们将讨论一些非角色特定的场景，可适用于任何构造块场景。这些可适用于一个分类或多个分类，原因是它提供了特殊的解析或可用性特征。

下面给出分类的构造块场景的概述。

（1）外部-外部分类。这个分类是由这样一个 DNS 部署组成的，其中要解析源于因特网的查询，即要查询您所在机构的公开（外部）解析信息。如果您有一条因特网连接，用于一个网站、电子邮件或其他公众可用的因特网应用，则在您的部署策略中就必须要处理这个分类。

1）外部 DNS 服务器部署。这个构造块场景寻求为外部客户端提供鲁棒的名字解析功能，这些客户端正在查找该组织机构的公开资源（例如 web 服务器、电子邮件服务器等类似资源）的合法名字解析，同时要最小化对如下客户端的暴露程度，这些客户端寻求攻击 DNS 基础设施或出于攻击目的而渗透到这些资源。外部 DNS 服务器的部署具有这样的特征，即一个隐藏的主服务器带有许多从属服务器。正如我们将看到的，这些服务器应该永远不会为一个解析器所直接查询；仅由递归名字服务器代表解析器进行解析。

（2）外部-内部分类。这个分类包括这样的查询，来自组织机构外部，寻找内部主机和资源的解析。除了为合作方提供“内部”解析信息的一个子集存取能力外，一般而言，应该禁用这个分类。这个分类的 DNS 服务器部署（用于合作方存取）应该模仿外部-外部分类下的外部 DNS 场景，但也许部署为一个并行的依据合作方不同的实现方法会更好。

（3）内部-外部分类。这个分类组成是，处理内部查询，它们请求因特网资源解析。

1）因特网缓存 *DNS* 服务器。因特网缓存服务器是内部 DNS 服务器，它们缓存因特网解析，由内部 DNS 服务器（代表的是内部解析器）所用。缓存服务器可被部署为内部解析 DNS 服务器的一项功能或独立于内部解析 DNS 服务器。在前一种情形中，

内部名字空间的权威服务器简单地将从因特网根服务器到域树的查询加速，以便解析查询，构造起被解析数据的一个缓存。后一种情形，使用独立的缓存服务器，支持其他内部解析 DNS 服务器配置成如下功能，即汇聚对外部数据的查询，通过这些缓存服务器实施解析。这样做的话，就支持对哪些服务器实施外部查询具有更多控制，而同时使这些服务器能够随时间推移构造起一个内容充实的缓存。

（4）内部-内部缓存。这个分类处理从内部发出的查询，这些查询要的是内部解析信息。

1）内部解析 DNS 服务器。要求 DNS 服务器解析来自内部主机对内部目的地的查询。这些 DNS 服务器配置带有内部名字空间的权威信息。任何因特网或不可解析的主机查询均可被汇聚到缓存（内部-外部）服务器。和外部主 DNS 服务器一样，出于增强安全和信息完整性的考虑，内部主 DNS 服务器应该是“隐藏的”。

2）部门级的 DNS 服务器。对于较大型的组织机构而言，一些商务部门或实体会希望在组织机构的名字空间内运行他们自己的名字子空间。这个场景的特征是在内部委派名字空间，但以另一种方式却是内部解析 DNS 服务器情形的一个复制，但也仅是内部名字空间的一个子集而已。

3）内部根服务器。内部根服务器可被配置为内部名字空间的权威根，用于内部查询的解析。

（5）通用交叉角色部署配置。这个分类可应用于多种部署场景。

1）秘密的从属 DNS 服务器。虽然在外部和内部部署场景中讨论了隐藏的主部署情形，但还有另一种“隐藏的”方法是隐藏从属服务器。一般而言，对于来自解析器以及其他名字服务器的直接查询，各主服务器是隐藏的；隐藏的从属服务器可用于解析器的名字解析，但对于来自其他名字服务器的请求迭代查询而言，却是隐藏的。

2）分割视图 DNS 服务器。一个域多个“版本”的部署方法，对于限制对特权的名字解析信息的存取而言，这种方法是有用的。BIND 9 的视图特征功能，支持一个域的多个视图或版本的部署。微软 DNS 目前不支持视图特征功能。

3）任意播服务器。任意播地址支持将单一 IP 地址指派到多台 DNS 服务器。这种做法支持使用一个共同的 IP 地址，将请求发送到该 IP 地址，目的是增强的性能和可用性。在网络中所用的路由协议实施这样的路由操作，即路由到带有任意播 IP 地址的最近距离的服务器。

依据您所在组织机构的规模（和预算），您可选择部署一组 DNS 服务器的外部-外部集合和一个内部-内部集合。这是最小的部署配置，但许多组织机构也部署专用的内部-外部服务器。更大型的或更复杂的部署方法可利用其他分类的组成单元。这种构造块方法的优势是简单性和模块化，使依据您的环境进行选择部署的场景成为可能。

11.3　外部-外部分类

这个分类与这样的部署有关，即对来自外部源的查询做出响应，这些查询需要的是该组织机构的公开解析信息。

11.3.1 外部 DNS 服务器

同样，“外部”指服务从组织机构之外或外部（即因特网）的 DNS 查询。必须为访问组织机构的网站、电子邮件和其他应用而提供解析服务，但在服务外部客户端中考虑到外部服务器的内在暴露程度和潜在脆弱性，为了保障这些外部服务器的信息完整性，必须谨慎从事。建议的方法是为服务外部请求，部署两台或多台从属 DNS 服务器，并为这些服务器配置 IPv4 和 IPv6 地址。这些从属服务器也许直接就部署在暴露于因特网的一个外部子网上，或位于一个 DMZ 内一个“一线”防火墙之后，如图 11-1 所示。

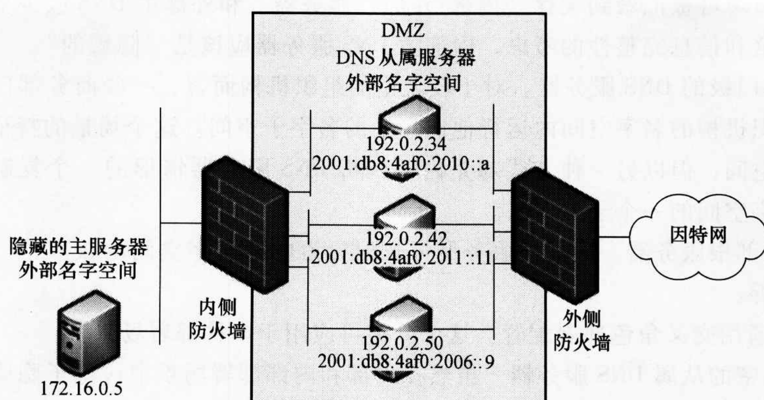


图 11-1 外部 DNS 构造块范例

我们将每台 DNS 服务器配置成双栈（IPv4 和 IPv6），以便支持通过任一协议的可达性。我们将每台外部 DNS 服务器放置在其自己的子网上，原因是我们为外部主机（例如这些 DNS 服务器）对我们的 192.0.2.0/24 进行分割。在这种情形中，我们形象地说明三个/29 子网，这些服务器部署在这些子网上：192.0.2.32/29、192.0.2.40/29 和 192.0.2.48/29。我们从三个/64 子网分配了三个 IP 地址，这些子网是从我们的 2001：DB8：4AF0：2000：：/56“外部”IPv6 分配中推算得到的。

图 11-1 形象地说明了一台隐藏的主 DNS 服务器，它部署在一个 DMZ 内部防火墙之后，不应由外部客户端直接查询。因为这台主服务器维护着“主配置”，从属服务器从这里得到配置信息，所以必须保障（safeguard）其信息的完整性。出于这个原因，这台主 DNS 服务器应该配置为隐藏的，这意味着它不能由查询其他 DNS 服务器的方法进行识别。隐藏主 DNS 服务器的方法，降低了一名攻击者识别该主服务器的风险，因为识别服务器之后，攻击者会尝试渗透其配置信息。想象一下这样的可能影响和尴尬境地，即如果一名攻击者将您的 www 记录更改为一个不正当的网站。隐藏一台主名字服务器的机制包括：在这个域的区域文件和父区域文件中去除该服务器的 NS 和黏结记录，并在每个区域 db 文件中修改 SOA 记录的主服务器名字（“mname”）字段。一般而言，面向外部的区域都是静态区域，是没有动态更新的，所以在这样的情形中可安全地实施 mname 字段修改。

确保配置在您所在父域中的 NS 和黏结记录，要指向外部从属 DNS 服务器，而不是主服务器。这应该通过您的 ISP 或域注册机构来安排组织。对于上图中的例子，您应该向您的父（例如 ISP）域管理员提供如下 NS/黏结记录信息：

ipamworldwide. com. 86400 IN NS extdns1. ipamworldwide. com.

ipamworldwide. com. 86400 IN NS extdns2. ipamworldwide. com.

ipamworldwide. com. 86400 IN NS extdns3. ipamworldwide. com.

extdns1. ipamworldwide. com. 86400 IN A 192. 0. 2. 34

86400 IN AAAA 2001：db8：4af0：2010：：a

extdns2. ipamworldwide. com. 86400 IN A 192. 0. 2. 42

86400 IN AAAA 2001：db8：4af0：2011：：11

extdns3. ipamworldwide. com. 86400 IN A 192. 0. 2. 50

86400 IN AAAA 2001：db8：4af0：2006：：9

注意，外部 DNS 服务器应该部署在不同子网和不同 ISP 连接上（如果可用的话），或使您的 ISP 也代表您运行一台从属服务器。我们可在我们的 ISP 连接防火墙上限制 DNS 查询，如表 11-1 和表 11-2 中的范例所示。出于简单性考虑，我们也将我们的三个/29 子网的规则合并成单一/27 网络。就防火墙配置而言，这些是简单的指导原则；您的策略可能是更加严格的。在 BIND 或 Windows DNS 中的类似 allow- * 选项设置（例如 allow-query（允许-查询）），可定义为针对每台服务器的访问控制列表（ACL）以及我们将在本章后面说明的那些方法。

正如刚刚提到的，为了最大化可用性，如果可能的话，这些服务器应该部署在多个位置。如果您有因特网双连接，则建议以一种类似配置在每个连接点处或附近部署外部从属服务器。图 11-2 形象地说明了一个多穴连接的外部 DNS 配置。在这种配置中，IPAM 全球公司的外部 DNS 服务器可通过 ISP、多条物理链路以及 DMZ 内部的不同子网均可访问到。

表 11-1　DNS 消息的外侧防火墙规则例

消息和方向	控制	源地址	源端口	目的地地址	目的地端口
来自因特网的 DNS 查询	Allow(允许)	Any(任意)	>1023	192. 0. 2. 32/27，2001：db8：4af0：2000/56	53
对 DNS 查询的响应	Allow	192. 0. 2. 32/27，2001：db8：4af0：2000/56	53	Any	>1023
所有其他情形	Deny(拒绝)	Any	Any	Any	Any

表 11-2　DNS 消息的内侧防火墙规则例

消息和方向	控制	源地址	源端口	目的地地址	目的地端口
由从属服务器到主服务器的查询(例如刷新查询)	Allow	192. 0. 2. 32/27	>1023	172. 16. 0. 5	53

（续）

消息和方向	控制	源地址	源端口	目的地地址	目的地端口
从主服务器到从属服务器的响应	Allow	172.16.0.5	53,1053	192.0.2.32/27	>1023
所有其他情形	Deny	Any	Any	Any	Any

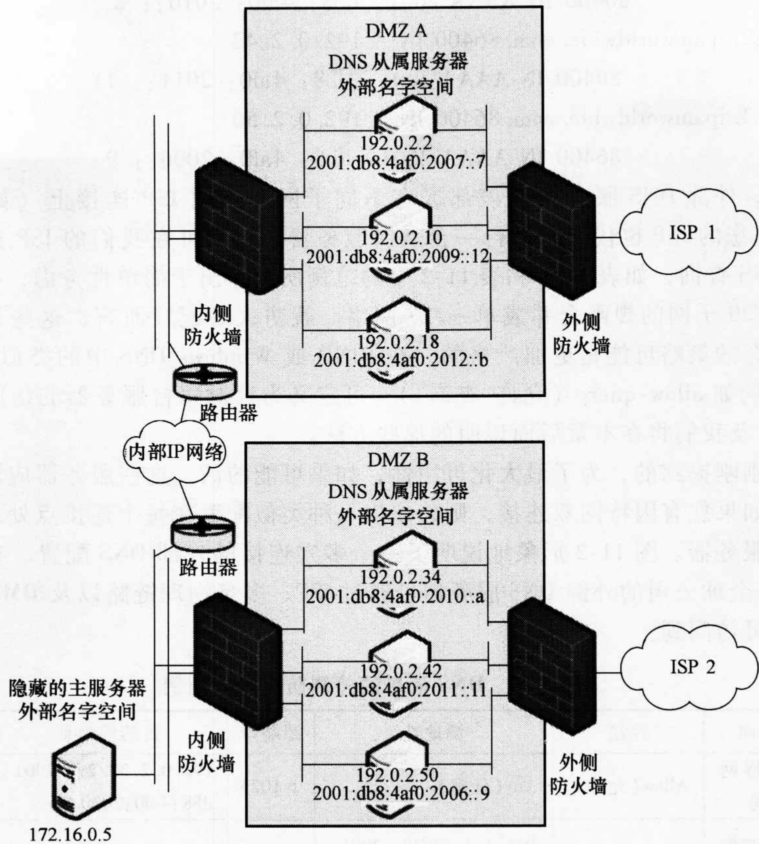


图 11-2 在一个多穴连接场景中的外部 DNS

注意到这些附加的建议，也将有助于维护外部 DNS 服务器以及它们所解析信息的安全。

（1）在一个 chroot 的监狱式法中允许 DNS。这降低了服务器平台及有关服务的暴露程度，可为其他攻击目标提供一个垫脚石，方法是将文件系统存取限制到一个有限的功能集合，而不是拥有对服务器上完全的根访问权限。这个“改变根权限”或 chroot 配置运行在某个名字的一个指定子目录下，而不是根目录。如果一名攻击者得到某个名字子目录的存取权限，他们将仅得到 chroot 后目录的存取权限，而不是根目录权限，在根目录权限下可存取所有的文件系统资源。多数工具性产品都预先配置，

在一个目录系统“jail”（监狱）中允许 DNS。

（2）跟踪使用 BIND 或微软 DNS 软件的最新版本，并订阅在 isc. org 或 Windows update（更新）的 bind-announce（绑定公告）电子邮件列表。一旦检测到安全弱点，就发送电子邮件通知，并带有关联的补救步骤和补丁。另外，监测针对平台所报告的操作系统弱点，这是因为您在这些平台上运行着 DNS。

（3）必须禁止递归查询。这将使这些服务器仅处理迭代查询，即来自其他 DNS 的查询，而不是解析器的查询。这些服务器一定不要代表任何人实施 DNS 查询。这就降低了支持递归查询的处理负载，更重要的是，降低了针对这些服务器的缓存毒化或拒绝服务攻击。

（4）在区域传递上配置访问控制列表，如我们将在下面说明的情形。

（5）如果有可能，就禁止针对这些外部区域的动态更新和通知。如果外部名字空间数据变化频繁，并要求动态更新的话，那么要限制主服务器及其从属服务器之间的更新和通知消息。

（6）为主服务器和从属服务器之间的区域传递和更新实施签名，而配置事务签名（TSIG）密钥，如我们将在下面说明的情形。这提供了数据源发者认证功能。

（7）如果需要通知，则在主服务器上配置端口号，在该端口号上要向从属服务器发送通知消息。这就要求在内部防火墙上指定相应的端口。

（8）在从属服务器上配置端口号，要在该端口上从主服务器得到区域传递。这就要求在内部防火墙上指定相应的端口。

（9）通过配置侦听地址/端口、允许来源和密钥语句，保障 rndc 控制信道的安全。您可能甚至想在这些服务器上禁止 rndc。

（10）将版本选项设置为一个伪造的设置。没有必要告知您正在运行的 BIND 的版本，因为这会为攻击者提供有关如何最佳攻击服务器的信息，特别当您没有保持使用较新的发行版软件时更是如此。

11.4　外部-内部分类

这个分类组成情况是，外部主机查询有关内部（非公开）解析信息的信息。一般而言，泄漏有关内部主机的信息是人们所不希望的，并是一项潜在的安全风险。即使相互连接的合作方仅应该访问受到保护的信息（当然不是整个的内部名字空间）时也是如此。

11.4.1　外部网 DNS 服务器部署

合作方间的连接，典型情况下，被配置为因特网之上的虚拟专网（VPN）连接或一个专网，典型地涉及合作方空间和内部网络之间的一个“合作方 DMZ”或防火墙。如图 11-3 所示，这个分类的 DNS 部署架构镜像了外部-外部分类的情况，但解析数据配置是多少有些不同的。取决于哪些解析数据可透露给一个给定的合作方，由合作方客户端查询的 DNS 服务器必须依据这种数据进行配置。因此按照外部 DNS 场景，不

支持递归的隐藏主服务器和可见从属服务器的概念是适用于这种分类的。

合作方特定的解析信息可被定义为一个“外部网”名字空间，其中包含配置于这些 DNS 服务器上的相应区域文件。另外，如果多个合作方存取访问一个共同的 DNS 服务器，服务合作方链路的 DNS 服务器上的实现视图，可支持每个合作方的解析信息。我们稍后将讨论 DNS 视图配置，但它们允许 DNS 服务器回答“谁在问”式的查询，例如合作方 A 的 ftp 主机名与合作方 B 的 ftp 主机名这两者的解析可能是不同的。

每个合作方的 DNS 解析过程应该被配置为可到达这些 DNS 服务器，以便解析您希望透露有关您所在网络的信息。我们将从内部-外部分类一节中的互补视角，配置我们的服务器来解析我们的合作方解析数据所需的信息。

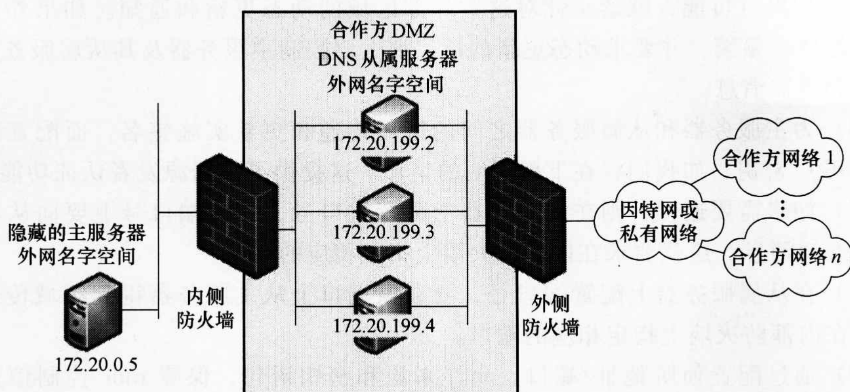


图 11-3　外部网 DNS 部署

11.5　内部-内部分类

11.5.1　内部解析 DNS 服务器

我们将回顾为内部解析部署 DNS 服务器的各种配置，包括各种主/从服务器配置、缓存服务器、外部网存取服务器和内部根服务器。

内部 DNS 服务器。应该部署内部 DNS 服务器，解析来自内部客户端对内部主机信息的查询。我们可称它们为内部 TLD，即“顶层域”，原因是它们将是内部名字空间的顶层主服务器（例如 ipamworldwide. com），并可将子域委派给其他内部 DNS 服务器，我们将在后面描述。内部主 DNS 服务器可被部署为一台隐藏主服务器。一般来说，通过隐藏主服务器而控制主服务器的信息完整性，是一个好想法，原因是内部发起的攻击占据网络安全泄漏事件的绝大部分。

部署足够数量的从属服务器（当然它们也是其相应区域信息的权威），可支持客户端查询的解析，同时卸载主服务器，使它们仅处理配置更新。如果一台主 DNS 服务器失效，从属服务器将继续解析查询，但一次长时间的中断可能破坏从属服务器区

域数据的有效性，当然就破坏了区域数据的及时性。从属服务器将继续支持这个区域数据，直到超过超时时间（expire time）为止，在此时间之后，该服务器将不再认为它自己是该区域的权威。如果主服务器下线，动态更新也就不可能了。

当尝试隐藏一台主 DNS 服务器时，这是客户端驱动的动态更新功能的微软客户端环境，需要考虑的一个重要因素是，微软客户端依赖于 MNAME 字段来识别要更新的主服务器。在这种情形中，使用 BIND DNS 服务器，您仍然能够隐藏主服务器，方法是将 MNAME 字段更改为指向一台合法的从属服务器，并配置 allow-update-forwarding 选项，将更新转发到主服务器。一般来说，我们建议，使客户端不能直接更新 DNS，而倾向于使您的 DHCP 服务器实施这项功能。能够更新 DNS 的实体越少，则可配置的访问安全就越严格，将能够影响 DNS 数据完整性的更新源种类就越少。

向笔记本计算机、台式机、打印机、VoIP 电话以及其他 IP 设备提供动态寻址功能，一般而言，使用 DHCP 服务器无论如何都是必要的。考虑到这些设备中的多数（如果不是所有的话）设备类型将要求在 DNS 中有对应于其相应指派地址的表项，我们就需要允许来自我们的 DHCP 服务器的 DNS 更新操作。因为我们有一台隐藏的服务器，所以我们能够配置 DHCP 服务器来更新一台从属 DNS 服务器。这台服务器可采用硬件冗余的方法进行部署，以便最小化任何中断服务时间，在这种情形中，DNS 不能为 DHCP 服务器实施更新。

图 11-4 给出一个范例，其中为我们的内部 ipamworldwide. com 名字空间部署了四台服务器。如在结构概述中所描述的情形，内部客户端解析器应该至少配置有两台 DNS 服务器。为了均衡查询负载，可在分支办事处或远端站点部署任何数量的附加从属服务器。

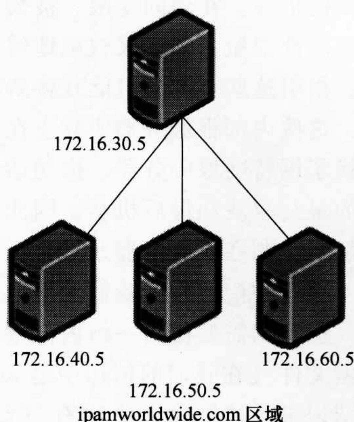


图 11-4　用于内部客户端的内部 DNS 服务器

11.5.2　内部委派 DNS 主/从服务器

在较大型的组织机构中，子域可被委派到特定的部门或分部。继续我们的范例，我们创建一个 finance. ipamworldwide. com 域，作为非委派的。这意味着，与 finance. ipamworldwide. com 域关联的配置和资源记录，被包括在其父区域文件中（ipamworldwide. com）。

对于其他部门，独立的 DNS 管理员可能希望管理他们自己的域信息。让我们考虑一个例子。如果工程部希望为 eng. ipamworldwide. com 域运行 DNS，则管理内部顶层域 ipamworldwide. com（即 eng. ipamworldwide. com 的父域）的团队可分配一个新的委派域（即区域）。这个新区域的权威 DNS 服务器的 NS 和黏结记录，需要在权威服务器自身和父区域 ipamworldwide. com 的那些权威服务器上进行配置。从技术角度而言，eng. ipamworldwide. com 区域（而不是其父区域）是这些 NS（和黏结）记录的权威，但父区域必须配置这些记录，以便提供沿域树向下的索引指示（referral）。

类似地，相关联的反向域（可能是多个）可在工程部门进行维护。让我们假定，它们从组织机构的私有空间分配了地址空间172.20.0.0/15。注意这个空间没有落在字节边界上。因此，将要求两个反向域。因为172.20.0.0/15由172.20.0.0/16和172.21.0.0/16地址空间组成，那么我们将创建两个基于字节边界的反向域：20.172.in-addr.arpa和21.172.in-addr.arpa。

图11-5所示为服务器部署范例。正如您看到的，它看起来极像带有一台主服务器和几台从属服务器的内部TLD DNS服务器部署情形。这是基于BIND的权威DNS服务器的常见部署配置。

图11-5　内部名字空间委派

11.5.3　内部根服务器

到此为止所描述的每台内部DNS服务器配置，都声明根（.）区域作为类型提示线索，指向root-hints.txt文件。到此时，我们假定这个文件包含因特网根服务器NS和A/AAAA记录，支持针对外部信息解析的域树规模扩展。在访问受限、被禁止或其他通常情况下因特网不可用的环境中，可部署一组内部根服务器，来权威地解析本地递归服务器不能解析的查询，或更可能的情况是，指引这些查询到可能获得解析的位置。

这些内部根服务器实际上在组织机构的其他服务器的hints（提示线索）文件中，替换了因特网根服务器。换句话说，即这些内部根服务器，是内部客户端索要名字解析的最终权威和最后机会。因此，这个配置消除了对因特网名字服务器的依赖，但却将解析限制在根服务器及其委派域服务器内所包含的信息上。

在这种配置中，我们使用上面针对我们的内部DNS服务器所定义的相同配置文件。但我们需要使用一组内部根服务器，而不是标准的因特网根服务器，来定义提示线索文件（在我们的情形中是root-hints.txt）。提示线索文件仅包含根服务器的NS和黏结记录。如果我们使用有三台根服务器的一个简单范例，则我们所参考（referenced）的提示线索文件也许看来像

```
.                           IN NS root1. ipamworldwide. com.
root1. ipamworldwide. com.  IN A 172. 16. 1. 1
                            IN AAAA 2001：db8：4af0：f1：：1
.                           IN NS root2. ipamworldwide. com.
root2. ipamworldwide. com.  IN A 172. 18. 1. 34
                            IN AAAA 2001：db8：4af0：a：：1
.                           IN NS root3. ipamworldwide. com.
root3. ipamworldwide. com   IN A 10. 251. 0. 5
                            IN AAAA 2001：db8：4af0：c001：：1
```

在每台根服务器上的配置文件也许看来像下面的情形：

```
aclinternal-nets {10. 0. 0. 0/8; 172. 16. 0. 0/12;2001:db8:4af0::/48; };
options {
   recursion no; // iterative queries only(仅支持迭代查询)
   allow-query { internal-nets; }; // allow from internal nets(允许查询来自内网)
   allow-notify { none; }; // disallow notify processing(不允许通知处理)
   allow-transfer { none; }; // disable zone transfers(禁止区域传递)
   allow-update { none; }; // disable updates(禁止更新)
};
zone ". " {
   type delegation-only;
   file "db. dot";
};
```

每台根服务器是一台仅支持委派类型的服务器，如在上面范例配置文件底部的根区域声明块内所指明的情况。仅支持委派类型是一种特殊形式的类型服务器，它仅以 referral（转荐）而不是答案做出应答。在 BIND 中多台主服务器配置的这样一种情形，对于像这种变化不频繁的静态区域是可能的。对根区域的任何修改，都意味着一个新的或修改过的顶层域指派，并必须通过更新每台根服务器上的 db. dot 文件来完成。没有动态更新、通知或区域传递。所有改变必须由管理员对 db. dot 文件的修改来完成，并要求所有主服务器上被修改区域文件的协同载入，才能将其同步地进行服务。

如下形象地说明了范例 db. dot 文件的一部分，它包含内部根区域的解析数据。

```
$ TTL 1d
. IN SOA dns1. ipamworldwide. com. dnsadmin. ipamworldwide. com (
        1 // serial number(序列号)
        2h // refresh interval of 2 hours(2h 的刷新间隔)
        30m // retry after 30 minutes(在 30min 后重试)
        1w // expire after 1 week(1 周后过期)
        1d ); // negative caching TTL of 1 day(1 天的负面缓存 TTL)
ipamworldwide. com. IN NS dns1. ipamworldwide. com.
                          IN NS dns2. ipamworldwide. com.
                          IN NS dns3. ipamworldwide. com.
                          IN NS dns4. ipamworldwide. com.
partner. net              IN NS dns-par1. ipamworldwide. com.
                          IN NS dns-par2. ipamworldwide. com.
...
16. 172. in-addr. arpa    IN NS dns1. ipamworldwide. com.
                          IN NS dns2. ipamworldwide. com.
```

```
...
0.f.a.4.8.b.d.0.1.0.0.2.ip6.arpa       IN NS dns1.ipamworldwide.com.
                                       IN NS dns2.ipamworldwide.com.
...
dns1.ipamworldwide.com.                IN A 172.16.40.5
dns2.ipamworldwide.com.                IN A 172.16.50.5
...
dns-par1.ipamworldwide.com.            IN A 172.20.199.2
dns-par2.ipamworldwide.com.            IN A 172.20.199.3
```

我们前面配置的对其他 DNS 服务器的转荐，支持查询的权威解析。这里，任何查询都落在 ipamworldwide.com 内（包括 eng.ipamworldwide.com），将被发送到我们的内部权威服务器。注意我们在这个列表中没有包括 172.16.30.5 这台服务器，原因是这是一台隐藏的服务器。

要求外部（例如）合作方外部网 DNS 服务器的任何解析，也都要求在提示线索文件中有对应的表项，它指向内部面向合作方的服务器。在我们上面的例子中，访问 partner.net 域及其子域，都将被转到权威 DNS 服务器 dns-par1.ipamworldwide.com 或 dns-par2.ipamworldwide.com。如我们将在下一分类中将讨论的情况，这些服务器可被配置为 partner.net 区域的桩服务器；另外，直接转荐到 partner.net DNS 服务器的方法，可被用于根区域文件内这些表项之上。底线是，这些根服务器可将顶层域委派给其他 DNS 服务器，这些服务器顺次被配置为：权威地解析相应的域和子域。

11.5.4 隐秘的从属 DNS 服务器

如此称为隐秘的（Stealth）从属 DNS 服务器，原因是在父区域中缺少服务器的 NS 和黏结记录，如我们在前一节刚刚看到的 172.16.30.5 服务器的情形；因此，这台隐藏的名字服务器不是通过 NS 查询加以识别的。我们已经使用这种配置来隐藏一台主服务器，但这种方法也可同样适用于从属服务器。因此，当遍历域树时，其他 DNS 服务器将不会因为解析而查询这台隐藏的服务器，原因是它没有在父区域的转荐中进行“通告”。

为了降低服务器间流量，或控制这种流量到解析器和其他服务器的一个固定组合，可针对一台从属服务器部署这种类型的配置。本地解析器可被配置为查询一台隐秘的从属服务器。并不采取去除隐秘从属服务器的 NS 和黏结记录，相反，该配置等价于一台正常从属服务器的配置。

11.5.5 多层服务器配置

部署隐藏服务器的方法，可有助于降低作为一组区域的主服务器对攻击的暴露程度。这使解析器和其他服务器可查询从属服务器，从属服务器也是所配置区域的权威。但在一些情形中，人们也期望增加一个第三层，以此补充两层的主从模型。这个高层的特征是一台主 DNS 服务，也许是所有内部名字空间的主服务器，实际上可提

供一个组织机构 DNS 信息的真正主数据库。这个场景如图 11-6 所示。

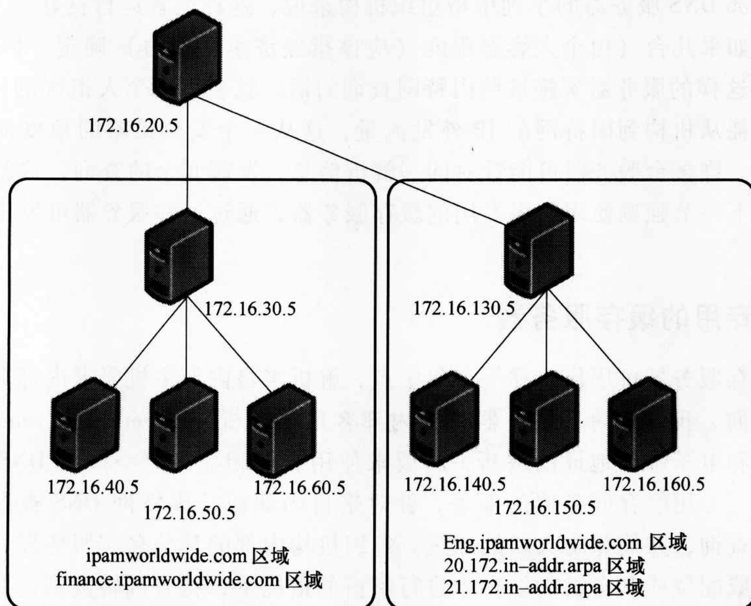


图 11-6 三层的内部服务器结构

让我们称这个顶层 DNS 服务器为一个层 1 服务器。它将所有区域配置为类型主模式（type master）。我们前面的主服务器 172.16.30.5 和 172.16.130.5（我们将称它们为层 2 服务器）现在被配置为从属服务器，从我们的层 1 主服务器处拉取区域传递。在层 3 的原始从属服务器集合，仍然保持为从属服务器，并继续从其相应的层 2 服务器拉取区域传递。这些层 2 服务器（虽然还是从属服务器）被配置在每台层 3 服务器区域语句的 masters 语句之内。因此，在我们的层 3 服务器的配置中不需要改变。但是，我们的层 2 服务器，必须针对每个配置的区域修改为从属服务器，以层 1 服务器标识为每个区域的主服务器。在这种配置中，层 1 服务器被称作主要的主服务器，原因是这是在其上区域更新可被直接执行的服务器，其中到层 2 和层 3 的区域传递被后续执行，以便据此更新所有的权威服务器。

11.6 内部-外部分类

部署场景的这个分类为组织机构内部的解析器，解决该机构外部信息的 DNS 解析。

11.6.1 混合权威/缓存 DNS 服务器

多数权威的、递归的 DNS 服务器，代表解析器，缓存查询解析过程中它们所接收到的解析信息，所以多数权威服务器从技术角度来说是“混合的”服务器。迄今为止我们讨论过的内部服务器配置都落在这个场景内：它们尝试解析权威信息，如果

不能解析，则逐步将查询升级，一直到因特网（或内部的）根服务器为止。对于拥有许多台内部 DNS 服务器的小到中型组织机构来说，这种配置运行良好。

但是，如果几台（由个人容忍程度（应该指经济承受能力）确定，但大约为 10 台或更多）这样的服务器实施这些因特网查询的话，就会出现令人担忧的问题。DNS 解析要求消耗从机构到因特网的 IP 外发流量，这从一个安全策略的角度而言，会增加暴露风险。许多台服务器可能针对同一解析信息，发起冗余的查询，这就降低了效率。因此，下一节强调使用一组专用的缓存服务器，通过这些服务器可发出所有的外发查询。

11.6.2 专用的缓存服务器

专用缓存服务器可用作这样一个集中点，解析来自内部主机要求内部名字空间之外信息的查询。我们的内部服务器将为内部客户端解析 ipamworldwide. com 查询，但因特网网站和电子邮件地址的解析，将要求使用支持相应名字空间的 DNS 服务器才能得到解析。专用缓存服务器的部署，针对来自内部源的因特网 DNS 查询，会有助于降低外发查询，并简化防火墙的配置。组织机构内部的其他名字服务器，在它们不能直接从权威配置或其自己的缓存中进行解析的情况下，将查询转发到这些缓存名字服务器。

由于对专用缓存服务器在代表内部主机解析因特网查询这方面的依赖，专用缓存服务器应该部署在一种高可用配置状态。因为这些缓存服务器将频繁地发送并接收因特网流量，所以它们应该被部署在靠近因特网连接处。将这种情形加到我们前面的外部 DNS 图中，得到图 11-7，它形象地说明了在内部网络内（但相对靠近因特网连接处）一对高可用服务器的部署。如果您拥有两条不同的因特网连接，则在靠近每条连接处部署一台服务器或一对服务器，是一个好想法，但在图 11-7 中仅给出一对服务器的情形。

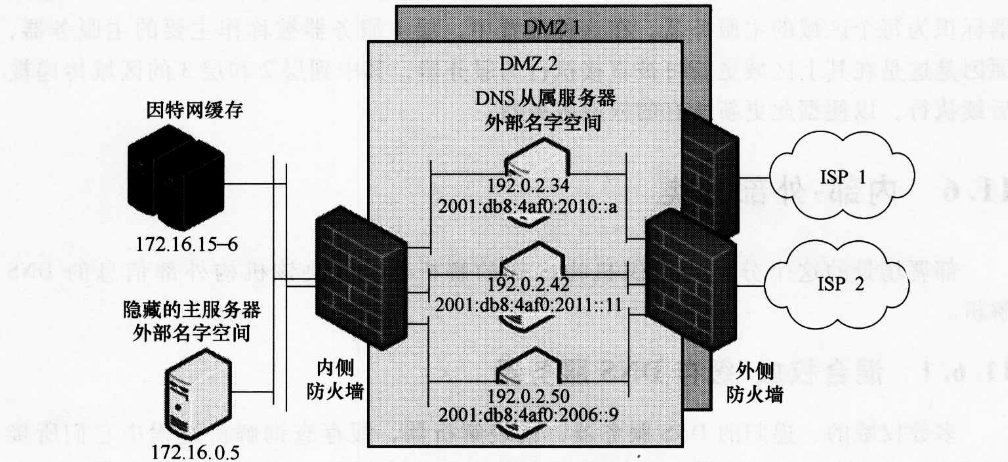


图 11-7 为外部解析，增加缓存服务器

外部服务器为外部查询器解析您的公开信息的查询，而缓存服务器代表您的内部客户端解析外部信息。内部缓存名字服务器的 IP 地址，需要添加到防火墙的允许列表，以便为内部客户端支持因特网主机名的解析。使用一台或少量这样的名字服务器，支持在满足如下条件的组织机构内部，仅指定这些少量地址而不是每台 DNS 服务器地址，否则该机构会执行迭代查询。

因为服务器为所有内部客户端缓存响应，所以随着时间推移，当服务器在其缓存中极可能有查询响应信息时，解析效率倾向于较高。但是，对缓存毒化的脆弱性以及不准确或恶意解析信息的使用，可能导致错误定向的应用连接。确保您的防火墙没有将外发 DNS 查询的 UDP 端口号做随机化处理。同样考虑在这些服务器上配置 DNS-SEC 验证选项（信任密钥），我们在第 13 章将讨论。我们前面讨论的防火墙范例配置可进行更新，见表 11-3 和表 11-4。我们使用“NAT {172.16.1.5}”来表示从缓存服务器的 IP 地址进行地址转换（NAT）得到的 IP 地址。针对每台服务器应该有这样的一个表项。

表 11-3　DNS 消息的范例更新外部防火墙配置

消息和方向	控制	源地址	源端口	目的地地址	目的地端口
从因特网来的 DNS 查询	允许	任意	>1023	192.0.2.32/27	53
对 DNS 查询的响应	允许	192.0.2.32/27	53	任意	>1023
因特网缓存服务器查询	允许	NAT{172.16.1.5}	>1023	任意	53
对因特网缓存服务器查询的响应	允许	任意	53	NAT{172.16.1.5}	>1023
所有其他情形	拒绝	任意	任意	任意	任意

表 11-4　针对 DNS 消息的范例更新内部防火墙配置

消息和方向	控制	源地址	源端口	目的地地址	目的地端口
由从属服务器到主服务器的查询（例如刷新查询）	允许	192.0.2.32/27	>1023	172.16.0.5	53
从主服务器到从属服务器的响应	允许	172.16.0.5	53,1053	192.0.2.32/27	>1023
因特网缓存服务器查询	允许	172.16.1.5	>1023	任意	53
对因特网缓存服务器查询的响应	允许	任意	53	172.16.1.5	>1023
所有其他情形	拒绝	任意	任意	任意	任意

缓存DNS服务器配置的重要方面包括如下内容。

1）作为一台缓存服务器，该服务器不是任何区域的权威。在这台服务器上要配置（或简单地包括）的唯一区域文件是root-hints. file。

2）仅允许来自内部源的查询。

3）禁止动态更新和区域传递。

4）可依据服务器资源和缓存策略，使用缓存管理选项。

5）当缓存服务器面临缓存毒化和其他“中间人”攻击时，采用安全补丁和更新使缓存服务器软件保持最新。在下一章讨论这点。

6）如果期望使用DNSSEC验证（在第13章详细描述），则要配置信任的或管理的密钥。

7）通过配置或缓存，如果其他内部DNS服务器不能解析的查询，它们会将这些查询转发到这些因特网缓存服务器。

（1）缓存服务器配置例。这种类型服务器的一个范例name. conf配置如下。

```
acl internal-nets { 10.0.0.0/8; 172.16.0.0/12;
        2001:db8:4af0::/48; } ;
options {
        directory "/opt/named/dns/etc";
        recursion yes;
        version "hidden";
        allow-query { internal-nets; };
        allow-transfer { none; };
        allow-update { none; };
};
zone "." {
        type hint;
        file "Internet-root-hints. file";
};
```

“Internet-root-hints. file”文件应该是标准的NIC根提示线索文件，指向因特网根DNS服务器。当这台服务器构建其缓存时，它将首先使用缓存来对内部查询做出响应，之后当必要时，从根服务器开始，沿域树向下进行查询。

（2）导致的内部服务器配置改变。考虑到将非权威解析上升到因特网（或内部）根，使用缓存服务器所导致的策略改变，对于到此为止我们所讨论的所有内部-内部分类服务器配置而言，需要做出如下配置改变。

1）去除指向提示线索文件的如下根区域语句块。

```
zone "." {
        type hint
        file root-hints. txt
}
```

2）激活到缓存服务器的转发功能，而不使用 hints（提示线索）文件。通过将如下语句插入到每台服务器的 named. conf 文件的选项｛｝块内，可做到这点。

```
options {
...
        forwarders { 172. 16. 1. 5 ; 172. 16. 1. 6 ;} ;
        forward only ;
...
}
```

这个配置指令 DNS 服务器将所有查询转发到我们的缓存服务器。转发选项可如上所述配置在全局层次、可配置在每区域层次以及可配置在带有区域层次的一个全局层次（例外的是在相应区域语句块内输入一个空的转发器语句（forwarders｛｝;））。

考虑我们的 172. 16. 40. 5 从属服务器配置，我们可更新它的 named. conf 文件，方法是如上所述去掉根区域块，并添加我们的两条转发语句。那么我们可使 ipamworldwide. com 区域免除转发，方法是在区域块内插入我们的空转发器语句，如下。

```
zone "ipamworldwide. com" {
      type slave ;
      forwarders {} ;
      masters { 172. 16. 30. 5  ; } ;
      file "bak. ipamworldwide. com" ;
} ;
```

采用这种配置，对 ipamworldwide. com 域内资源记录的查询将由服务器自己来解析；其他查询将被转发到我们的因特网缓存服务器。

这个配置的另外一种方法涉及在每台服务器上保留根区域和提示线索文件配置，同时要将 named. conf 文件内 "forward only;" 语句改变为 "forward first;"。这样做，就将服务器配置成：在开始时尝试使用转发来解析查询，但如果不能解析，则使用其他的方法，例如通过根服务器的解析法。"forward only;" 语句指令服务器转发，以此作为解析的唯一（first and last resort）选项。

11. 6. 3 外网解析服务器

这个场景包括：配置内部 DNS 服务器解析来自内部客户端请求外网合作方信息的查询。这种配置是我们在外部-内部分类中所讨论外网场景的补充。在这种特定的场景中，存在三种配置可能性。

（1）外网转发区域。在前一节我们对转发的讨论之上，我们可将与合作方域绑定的查询定义为类型为"转发"的一个区域。内部 DNS 服务器所接收到的、要求对合作方域解析有关的所有查询，将被转发到指定的合作方 DNS 服务器。为了针对我们的合作方域（比如 parner. net）实现这个场景，我们在 named. conf 文件内声明这个区域，如下。

```
zone "partner. net" {
```

```
        type forward;
        forwarders { 192.168.100.5; 192.168.200.5; };
        forward only;
};
```

（2）外网桩区域。一个桩区域是从属区域的一种特殊形式，其中在服务器上仅传递和维护该区域的NS和黏结记录，而不是整个区域资源记录内容。就像一个桩解析器一样，一个桩区域是这样配置的，就给定区域为各查询配置“谁在请求”（who to ask），而不像一个从属区域那样直接提供答案。在一条外网链路的特殊情形中，可为我们的合作方域partner.net，产生一个桩域。

```
zone "partner.net" {
        type stub;
        masters { 192.168.249.11; };
};
```

（3）通过内部根的外网委派。如果您正在使用内部根服务器，则您可将partner.net区域定义为仅是委派的，并指向合作方的DNS服务器（可能是多台服务器），方法是配置相应的NS和黏结记录。在我们前面的内部根区域文件例子中，我们将要解析查询的DNS服务器内部集合指向partner.net域。这些被索引（referenced）服务器将需要配置这个区域为转发或桩；另外，根区域文件可直接指向合作方的服务器，虽然这样做时，如果您的合作方改变服务器IP地址或主机名的话，这就会成为一个维护问题。

11.7 交叉角色分类

11.7.1 分割视图DNS服务器

BIND 9的这种视图特征功能支持将一个区域的多个版本部署到一台DNS服务器上。可依据查询源和目的地上的一个地址匹配列表以及查询是否为递归的，服务器可过滤查询。这个过滤过程确定为查询的解析，将搜索区域的哪个版本或视图。每个视图均被配置有其区域文件的对应版本。使用视图的一个常见例子是用于“分割DNS”，或提供一个组织机构名字空间的内部和外部版本。虽然并不建议部署单一DNS服务器集合来处理内部和外部查询，但对于较小型的组织机构，这确是一种更实际的方法。但视图也可用于内部名字空间，将对某些主机解析的访问限制到特定的解析器客户端。

在下面的例子中，我们将定义一个新的子域hr.ipamworldwide.com，定义该区域的两个版本，之后将这些版本与DNS服务器上的相应视图关联。这个概念是提供对hr（人力资源）门户服务器（主机名portal）的通用访问，但将hr.ipamworldwide.com子域内其他主机的访问限制到仅有网络上的HR用户才有权限。

我们首先考虑的一件事是区域文件自身。让我们在一个称为

db. hr. ipamworldwide. com. default 中创建我们的通用的或默认的域版本，在另一个称为 db. hr. ipamworldwide. com. hr 的文件也做这些工作，但后者对于 HR 客户端具有较宽的可见性（visibility）。注意如果没有委派 hr 子域，则将需要父区域文件的两个版本，子域的每个视图需要一个版本。

我们的 db. hr. ipamworldwide. com. default 文件将包含有限数量的 A 记录或甚至仅有一条 A 记录（除了该区域的 SOA、NS 和黏结记录外）。

```
portal IN A 172.16.4.24
```

而 db. hr. ipamworldwide. com. hr 文件将包含更多的资源记录，例如

```
payroll IN A 172.16.4.10
benefits IN A 172.16.4.14
empdb IN A 172.16.4.22
portal IN A 172.16.4.24
promo IN A 172.16.4.30
```

注意也需要 4. 16. 172. in-addr. arpa. 域的两个版本，每个版本对应于每个视图中暴露出的 IP 地址和主机。在 db 文件名上，我们使用相同的 default 和 hr 后缀。

db. 172. 16. 4. default 文件概要：

```
...
24 IN PTR portal.hr.ipamworldwide.com.
...
```

db. 172. 16. 4. hr 文件概要：

```
...
10 IN PTR payroll.hr.ipamworldwide.com.
14 IN PTR benefits.hr.ipamworldwide.com.
22 IN PTR empdb.hr.ipamworldwide.com.
```

244 DNS SERVER DEPLOYMENT STRATEGIES

```
24 IN PTR portal.hr.ipamworldwide.com.
30 IN PTR promo.hr.ipamworldwide.com.
...
```

既然我们已经为每个版本创建了我们的区域文件，那么我们可将它们与 DNS 服务器上的对应视图关联。让我们使用 hrdns. hr. ipamworldwide. com 作为我们的主 DNS 服务器。下面是这台服务器的范例 name. conf 文件的部分内容：

```
acl human-res {172.16.4.0/23; };
view "hr" {
    match-clients { "human-res"; };
     match-destinations { localnets; };
     zone "hr.ipamworldwide.com." {
          type master;
          file "db.eng.ipamworldwide.com.hr";
```

```
        };
        zone "4. 16. 172. in-addr. arpa." {
            type master;
            file "db. 172. 16. 4. hr";
        };
};
view "default" {
        match-clients { any; };
        zone "hr. ipamworldwide. com. " {
            type master;
            file "db. hr. ipamworldwide. com. default";
        };
        zone "4. 16. 172. in-addr. arpa. " {
            type master;
            file "db. 172. 16. 4. default";
        };
};
```

注意在 named. conf 内视图语句的顺序是重要的。匹配查询的第一个视图将被用于确定将访问哪个区域文件的信息。因此我们在更具区分力的 hr 视图语句之后定义我们的默认视图，它匹配所有的客户端查询。我们声称这是一个部分定义。为了合适地实施到从属服务器的区域传递，要以相同方式（likewise）配置这些视图，要求进行特殊处理。毕竟对与 hr. ipamworldwide. com 主机有关的一条资源记录的更新，可解释为落入任一视图。应该更新哪个视图呢？

要确保通知、更新和传递处于正确的视图间，就要求对匹配客户端 ACL 以及查询-源、通知-源和传递-源语句进行修改（manipulation）。通过为这样的每个功能定义源 IP 地址、施用一条对应的 ACL，服务器间通信可被集中到正确的视图。在对我们上述例子扩展上，我们需要修改我们的 “human-res”（人力资源）ACL 以便禁止（negate）默认视图的源语句中使用的 IP 地址（可能有多个地址）。即考虑到顺序的重要性，来自于主服务器对默认视图的通知，需要从人力资源的视图中阻塞掉，使他们看不到。类似地，来自于从属服务器针对默认视图的传递请求，也需要从人力资源的视图中阻塞掉。另外，在两组服务器上的查询-源应该进行类似配置。

这在图 11-8 中作了形象的说明。出于简洁性考虑，在每台服务器上使用字母来符号化（symbolize）*-source[⊖]选项，我们可以图形方式说明，有关视图 2 的一个区域更新的一条通知（从主服务器到从属服务器 1）将首先匹配从属服务器上的视图 1。来自视图 2 的通知，将使用源 IP 地址 B。在从属服务器 1 上，在视图定义的匹配-

---

⊖ 即在主服务器上的查询-源（query-source）和通知-源（notify-source）选项以及每台从属服务器上的查询-源（query-source）和传递-源（transfer-source）选项。

客户端部分内阻塞 B，但落在视图 2 的匹配准则内，因此得以施用。类似地，来自源在 IP 地址 I 上针对默认视图的、由从属服务器 2 到主服务器的一条传递请求，将在视图 1 和视图 2 内被阻塞，但正确地施用到默认视图。

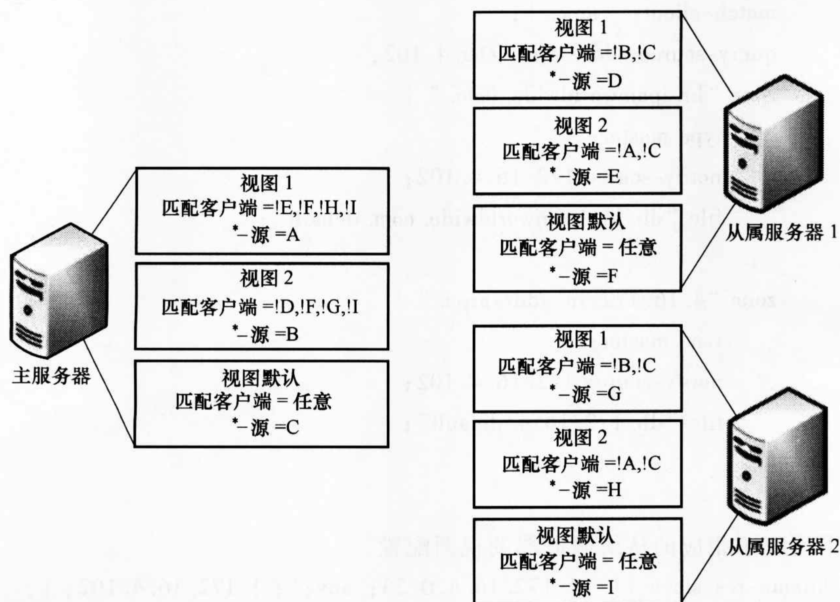


图 11-8　服务器间通信的视图配置

将这个模型施用到我们上面较简单的配置例，其中一台主名字服务器使用 IP 源地址 172. 16. 4. 101（见图 11-8 中的地址 A）和 172. 16. 4. 102（地址 C）、单一一台从属服务器施用源地址 172. 16. 5. 201（地址 D）和 172. 16. 5. 202（地址 F），我们得到主服务器的如下配置。

```
acl human-res { ! { ! 172. 16. 4. 0/23; any; }; ! 172. 16. 5. 202; };
view "hr" {
        match-clients { "human-res"; };
         match-destinations { localnets; };
         query-source address 172. 16. 4. 101;
         zone "hr. ipamworldwide. com. " {
              type master;
              notify-source 172. 16. 4. 101;
              file "db. eng. ipamworldwide. com. hr";
         };
         zone "4. 16. 172. in-addr. arpa. " {
              type master;
              notify-source 172. 16. 4. 101;
              file "db. 172. 16. 4. hr";
```

```
        };
};
view "default" {
        match-clients { any; };
        query-source address 172.16.4.102;
        zone "hr.ipamworldwide.com." {
            type master;
            notify-source 172.16.4.102;
            file "db.hr.ipamworldwide.com.default";
        };
        zone "4.16.172.in-addr.arpa." {
            type master;
            notify-source 172.16.4.102;
            file "db.172.16.4.default";
        };
};
```

如下反映了相应的从属服务器的视图配置。

```
acl human-res-slave { ! { ! 172.16.4.0/23; any;}; ! 172.16.4.102; };
options {
        directory "/opt/dns/etc";
};
view "hr" {
        match-clients { "human-res-slave"; };
        match-destinations { localnets; };
            query-source address 172.16.5.201;
            zone "hr.ipamworldwide.com." {
                type master;
                notify-source 172.16.5.201;
                masters { 172.16.4.101; };
            };
            zone "4.16.172.in-addr.arpa." {
                type master;
                notify-source 172.16.5.201;
                masters { 172.16.4.101; };
            };
};
view "default" {
        match-clients { any; };
```

```
        query-source address 172.16.5.202;
        zone "hr.ipamworldwide.com." {
                type master;
                notify-source 172.16.5.202;
                masters { 172.16.4.102; };
        };
        zone "4.16.172.in-addr.arpa." {
                type master;
                notify-source 172.16.5.202;
                masters { 172.16.4.102; };
        };
    };
```

11.7.2 采用任意播地址部署 DNS 服务器

采用任意播地址配置 DNS 服务器，支持多台 DNS 服务器使用同一个 IP 地址。回顾一下，一个任意播地址是指派到多个接口（典型情况下在不同节点上）的一个地址。当尝试到达任意播可寻址主机中的任意一台时，如果不关心到达的是哪台主机，则使用任意播。路由设施处理路由度量指标的更新，以便跟踪到配置有目的任意播地址的最近主机的可达性和路由。图 11-9 形象地说明了这样一个例子，其中有三台 DNS 服务器，配置有任意播地址 10.4.23.1。

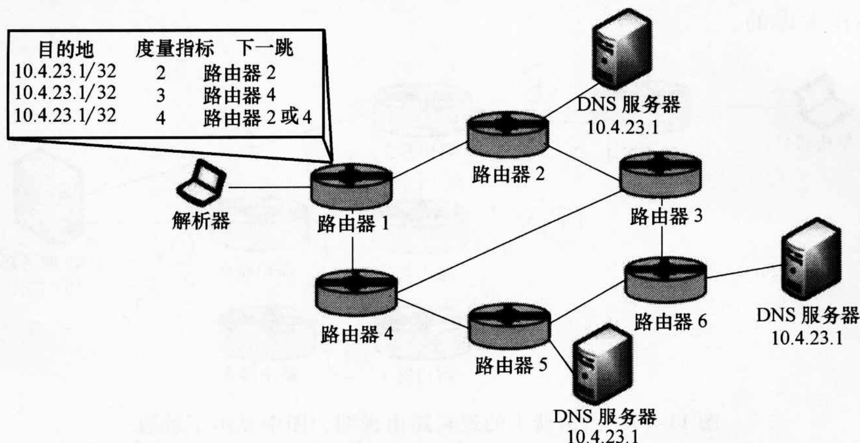


图 11-9　任意播路由表例

如图 11-9 所示，路由器 1 有到任意播地址 10.4.23.1/32 的三条路由，它们对应于我们的三台服务器。最近的服务器是驻留（home）在路由器 2 上的，距离路由器 1 有两跳。下一个距离最近的服务器驻留在路由器 5 上，通过路由器 4 在三跳内可达。最后，连接到路由器 6 的服务器，通过路由器 2 或路由器 4 在四跳内可达。从路由器 1 视角看的逻辑视图如图 11-10 所示，其中任意播 IP 地址被看做单一目的地，通过多条路径可达。

（1）任意播的优势。部署任意播的做法，提供了诸多优势。

1）简化了解析器配置。

2）改善的解析性能。

3）高可用性的 DNS 服务。

4）对 DNS 拒绝服务攻击的抑制能力。

配置有 DNS 服务器任意播地址的各解析器，它们的查询将被路由到配置有那个任意播地址的最近 DNS 服务器。因此，不管解析器主机连接到哪里的网络，相同的任意播 IP 地址都由解析器使用来定位一台 DNS 服务器。这个局部化的查询过程也被用来改进解析过程的性能。到一个 DNS 任意播地址的一条查询，应该被路由到最近的 DNS 服务器，由此降低了整体查询过程中的往返延迟时间量。

为了相应地更新路由表，一台 DNS 服务器的连接中断可通知到路由基础设施（没有通信量就表示中断了）。这要求 DNS 服务器运行一个路由守护进程，使用所选的路由协议将可达性信息通知给本地路由器。参与到路由协议更新之中的做法，支持本地路由器以一个合适的度量指标更新它的路由表，并通过路由协议将这个信息传递给其他路由器。取决于 DNS 服务器的部署情形（内部的或外部的），将需要在 DNS 服务器上运行一个相应的内部或外部路由协议。服务器简单地需要通知它的任意播地址是可达的。典型情况下，是这样完成的，将服务器本地回环地址[⊖]之一指派为任意播地址，并在一个或多个端口上运行一个路由守护进程，通告到该任意播地址的可达性。在如下情况下，这将是特别有用的，即如果这个路由更新被连接到服务器上的 DNS 守护进程或服务器的状态时，虽然当通知 IP 地址可达性时，一般而言，应用状态是不作考虑的。

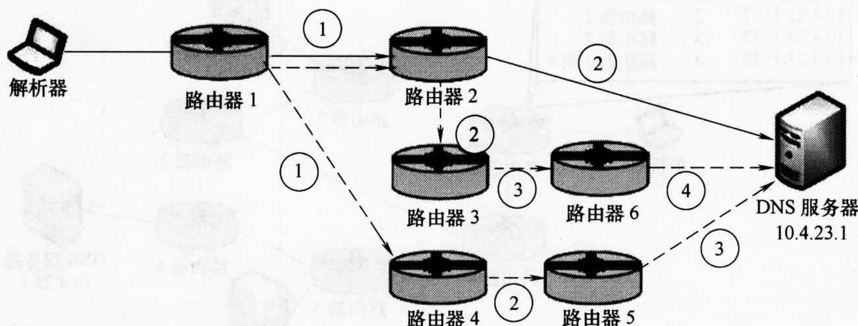


图 11-10 路由器 1 的逻辑路由视图，图中显示了跳数

部署任意播，要付出拒绝服务攻击的缓解措施的代价，这已被 2007 年 2 月 6 日对多台根服务器的分布式拒绝服务（DDoS）攻击所验证。在作为攻击目标的六台根服务器中，受影响最严重的两台是还没有实施任意播的那些服务器。部署了任意播的其他四台根服务器，支持将攻击分散到更多物理服务器的能力。因此，对 I 根服务器

⊖ 这里的术语“回环地址”指代软件回环地址，普遍实现为路由器和服务器中的“设备地址”，在这些设备的任意接口上均可达。

（还没有实施任意播）的一次 DDoS 攻击，严重地影响了服务器对合法查询响应的能力，而对 F 根服务器（它在 40 台以上的服务器上配置了 F 根任意播地址）的攻击，被服务器将攻击的影响分布到了这些服务器上。这种形式的负载分担做法，使 F 根服务器（可能是多台）在遇到人为请求的密集攻击时，继续处理合法的查询。

（2）任意播警告（caveat）。虽然任意播提供了许多优势，但也要考虑到部署任意播的约束和警告。因为各解析器可在一个给定的时间查询配置有任意播地址的任何一台 DNS 服务器，所以配置在服务器上的解析信息的一致性，是重要的。例如，在因特网根服务器上的实现，由带有静态信息的一组主服务器组成。这些根服务器不会接受动态更新。如果动态区域希望使用任意播，那么除了它的任意播地址之外，每台服务器必须有一个单播地址[⊖]。这种做法支持将更新定向到主服务器的单播地址，接下来通过从属服务器相应的单播地址来通知主服务器的从属服务器。可使用一种隐藏的主服务器配置，它的从属服务器配置有任意播地址。

另一种考虑是要求在您的 DNS 服务器上运行一个路由守护进程，该服务器配置有任意播地址。将报文路由到任意播地址主要（primarily）是一项路由功能，一台 DNS 服务器主机的不可达会导致查询尝试的丢失。在如下情况下将会发生这种情形，即静态路由被用来配置带有 DNS 服务器的固定度量指标的路由器，这些 DNS 服务器配置有一个共同的任意播地址。如果一台服务器变得不可达，则服务路由器就没有办法检测到这点，将不会重新路由目的地为任意播地址的报文。因此，在 DNS 服务器上集成一个路由守护进程，就改进了整体的鲁棒性。如果一台服务器出现故障，则本地路由器将确定它不再是可达的，并将更新它的路由表，通过路由协议更新操作来更新其他路由器的路由表。考虑到因特网根路由器部署在全球因特网上，它们支持 BGP，但部署在组织机构内部的将可能要求支持 OSPF、IGRP 或所选中的内部路由协议。

最后，当使用任意播时，排错是有点挑战性的。考虑到服务器的二义性，要对来自一台主机任意播地址的伪造响应进行排错，是困难的。为了识别哪台以任意播寻址的服务器是有问题的，一种好的思路是，以 BIND server-id 选项配置服务器身份识别。您可定义一个字符串标识符或仅使用主机名参数，来使用服务器的主机名。通过发出带有 qname =” ID. SERVER”、qtype = TXT（qclass = CHAOS）的一条查询，可检索这个值。使用这项挖掘（dig）设施工具，这看起来像

```
dig id. server chaos txt@ <anycast-address>
```

也许一种更好的方式，是对一条查询使用 dig +nsid 参数，所以您可在一次事务中将一条不良的响应与服务器身份识别相关。

```
dig +nsid <query>@ <anycast-address>
```

要了解有关任意播配置的更多细节，请见参考文献［137，138］。

⊖ 出于管理目的，每台任意播服务器将要求一个单播地址，但为了支持动态区域，要求为一个接口提供一个附加的单播地址，用于更新、通知和区域传递。

11.8 将所有情况整合起来

在本章我们给出许多构造块场景，解决处理各种配置，每种配置都将目标锁定在解决一个特定的 DNS 解析目的。取决于您所在 IP 网络的尺寸和规模，您可选择为您网络上的不同应用实现几种构造块场景。仅需要记住，不存在真正的曲奇饼成型刀式的答案；这些场景中的每种场景都应该依据您的个体需求而进行评估。这里的目的是以一种模块化的形式提供一些指南，以便帮助简化整体的部署设计过程。

我们有意地定义这些场景构造块中的每个构造块，它们具有其自己的离散 DNS 服务器集合。这种基于角色的方法有助于模块化，但也有助于排错和管理安全策略。比如使用同一台服务器处理内部和外部解析，这是一种方法，与此方法不同，将这些功能隔离在多台服务器间的做法，则提供了物理上的和功能上的隔离。

多数组织机构将部署最小化的外部 DNS 场景（外部-外部分类）和内部 DNS（内部-内部）构造块。拥有合作方的那些组织机构也添加外部网场景，以便涵盖进入的和外发的查询解析。因特网缓存服务器可以这样部署，即支持外发因特网解析的汇聚集中，并随时间推移构建形成丰富的缓存信息。可依据您的需要，也可配置视图和/或任意播。场景组合的潜在数量是无穷尽的。图 11-11 形象地展示了 IPAM 全球公司整体 DNS 基础设施的一种可能状态。

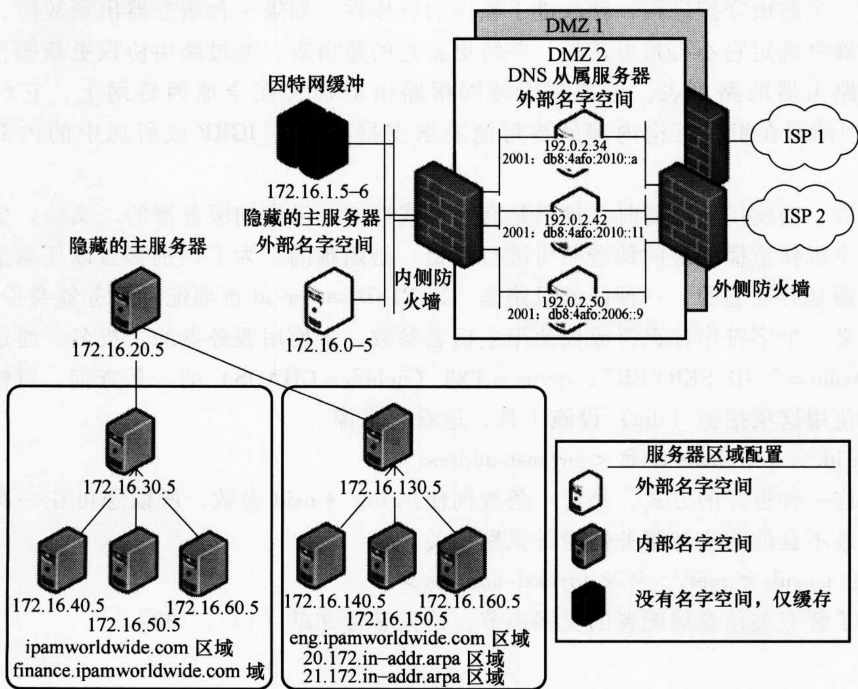


图 11-11 IPAM 全球公司的 DNS 服务器部署

第12章　保障DNS安全（上）

12.1　DNS弱点

如我们已经看到的，DNS是几乎每项IP网络应用可用性的基础，这些应用从网页浏览、电子邮件直至多媒体应用等。使DNS服务不可用的一次攻击，或攻击对包含于DNS内部数据完整性的修改，实际上这些可能造成一项应用或网络的不可达。明显的是，在整个解析过程中，对DNS数据和DNS通信的保护，是至关重要的。本章将焦点放在从总体角度来看DNS内部的潜在安全弱点上。特定的DNS服务器实现可能包含其他弱点，例如与任何网络服务器或应用、操作系统监测及关联的软件弱点，这是一个基本运行过程。

在讨论DNS安全弱点时，考虑DNS内的数据源和数据流，是有启发指导意义的，如图12-1所示。从图的右上角开始，DNS服务器最初被配有配置和区域文件信息。这个配置步骤可使用一个文本编辑器或一个IPAM系统来实施。对于微软的实现而言，“IPAM系统”就是微软管理控制台（MMC）。要求进行服务器参数和相关区域的配置。

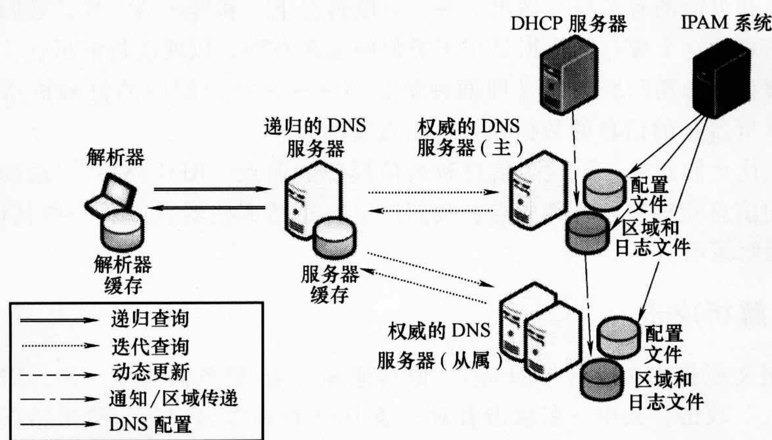


图12-1　DNS数据存储和更新源[11]

对于BIND实现而言，这个配置由主服务器上一个named.conf文件和关联的区域文件组成。虽然一台服务器可能是一些区域的主服务器和其他区域的从属服务器，但我们将使用主服务器术语（terminology）来估计一个特定区域有关信息的弱点。从属服务器的配置仅要求创建配置文件，它定义了服务器的配置参数及其特定区域的权威权限。从属服务器将区域信息从相应主服务器传递过来。

区域信息也可由外部源进行更新，特别是由 DHCP 服务器进行更新。动态更新可为各客户端接受，这些客户端得到动态 IP 地址，这些 IP 地址要求进行地址到名字映射的 DNS 更新。典型情况下，这些更新源于 DHCP 服务器指派地址的过程，并将被定向到作为给定区域主服务器的服务器。主服务器将该更新添加到它的日志文件，之后会将更新通知它的从属服务器，从属服务器可能请求一次增量式的区域传递，以便捕获更新过的区域信息。

因此，直接通过区域文件编辑或由一个 IPAM 系统，并通过区域传递和动态更新的方法，可在一台名字服务器上配置权威区域信息，得到几个潜在的数据源和数据更新通信路径。

除了配置信息和区域文件外，一台 DNS 服务器内的第三个信息库是它的缓存。通过一个查询解析过程，积累得到缓存信息。当查询答案被查找和接收时，相应的答案就被服务器缓存。被缓存的信息不仅可从 DNS 协议消息的答案节得到，而且可从权威节和附加节得到。据称，这些节提供权威服务器信息和对答案提供补充的信息。这个信息可包括相关区域的权威服务器以及与查询有关的其他信息（例如一条 NS 查询的 A/AAAA“黏结”记录）。

在图 12-1 节左侧开始的查询解析流，开始时，客户端解析器向其递归服务器发起一条递归查询。回顾一下，要查询的目标服务器，是定义在客户端的解析器配置中的、人工管理或通过 DHCP 管理的。必要情况下，递归服务器通过域树将发出迭代查询来解析查询，一般情况下，终止于一台服务器，它是对应于该查询区域的权威。主服务器或任何从属服务器是带有区域信息的权威服务器。权威服务器在响应的附加节中，以答案和可能的有关信息做出应答。一般情况下，和解析器一样，递归服务器将缓存这个信息。这个缓存的依据是用于类似的未来查询，以便改进解析性能。所以确保返回给解析器和递归服务器（即两种解析器——在客户端中的桩解析器和递归服务器内的解析器）的信息的数据完整性是重要的。

现在，让我们研究一下这个信息和通信模型的弱点。RFC 3833[139] 透彻地讨论了 DNS 协议和信息完整性的各种弱点。我们这里将总结那些弱点以及一些其他的弱点，之后研讨缓解策略。

12.1.1 解析攻击

（1）报文截获或伪造（spoofing）。像其他客户端/服务器应用一样，DNS 容易受到“中间人”攻击，其中一名攻击者对一条 DNS 查询做出响应，给出错误的或误导的附加信息。攻击者伪造 DNS 服务器响应，引导客户端来解析并缓存这个信息。这可能导致被劫持的解析器因此劫持应用到不正确的目的地（例如网站）。

（2）ID 猜测或查询预测。另一种形式的恶意解析是 ID 猜测。和在 UDP 报文首部 ID 一样，DNS 报文首部的 ID 字段是 16bit 长的。如果一名攻击者可提供带有正确 ID 字段和 UDP 端口号的一条响应，则解析器将接受该响应。这会使攻击者能够提供伪造的结果，其中假定的是攻击者已知或猜测到查询类型、类和名字。这种攻击可潜在地将主机重定向到一个违法的站点。即使采用蛮力攻击方法，猜测一个 2^{32} 数也是

相对容易的。

（3）名字链或缓存毒化。这种报文截获类型攻击的特征是，一名攻击者通常在 DNS 响应报文的附加节或甚至权威（Authority）节提供补充解析信息，因此就以恶意的查询信息对缓存实施了毒化。这可能是（例如）尝试伪造一个流行的网站（例如 cnn. com、google. com 或类似域名），所以当请求这样的一条查询时，解析器将依赖于这种伪造的缓存信息。当解析器被请求解析这样的一条查询时，它将访问其缓存，并利用恶意信息，本质上会将客户端重定向到攻击者设计的（intended）目的地。强迫解析器访问被毒化缓存数据的另一种方法，是提供一个电子邮件链接，当单击该链接时，将会解析到设计好的被毒化的主机名。所谓的 Kaminsky DNS 弱点就是一种缓存毒化的攻击类型。

（4）解析器配置攻击。在客户端上的解析器必须被配置至少一个 DNS 服务器 IP 地址，可向该地址发出 DNS 查询。这种配置可按如下进行，人工地将 DNS 服务器 IP 地址硬编码到 TCP/IP 的协议栈，或通过 DHCP 或 PPP 自动地得到。这种类型的攻击也可能源于一名攻击者，例如，它通过一个 web 插件发起的。这种类型的攻击寻求将解析器重定向到一名攻击者的 DNS 服务器来解析到恶意数据。

12.1.2 配置攻击和服务器攻击

（1）动态更新。通过尝试到服务器的一条动态更新，一名攻击者可能尝试向一个 DNS 区域中注入数据或修改数据。这种类型的攻击，尝试将来自客户端查询预期目的地的解析，重定向到一个攻击者指定的目的地。

（2）区域传递。冒充一台从属服务器，并尝试实施从一台主服务器的一次区域传递，这种做法是这样一种形式的攻击，它尝试映射区域或对区域做上踪迹。即通过识别主机到 IP 地址的映射以及其他资源记录，攻击者尝试识别可直接攻击的目标。使用主机名作为一个提示信息（例如“工资表”（payroll）），来攻击一台特定主机，提供了尝试访问服务或拒绝服务（DOS）的一个简易目标。

（3）服务器配置。一名攻击者会尝试得到运行 DNS 服务器的物理服务器的访问权限。在本地或远程访问服务器时，要求提供一次登录和口令，对于对抗直接服务器访问，高度推荐使用这种做法。对于远程访问，也推荐使用安全外壳（SSH）。当使用一个 IPAM 系统时，验证 IPAM 到 DNS 服务器间的通信是安全的。除了能够篡改命名和区域信息外，这种类型的攻击当然可使用服务器作为其他目标的一块垫脚石，特别当这台服务器恰巧为内部所信任时尤其如此。

（4）控制通道攻击。对 ndc 或 rndc 通道的访问，提供了强大的远程控制能力，例如停止/终止进程（named）、重新调入一个区域等。访问控制通道，并停止服务，因此就拒绝了对查询服务器和解析器的服务。

（5）缓冲溢出和操作系统攻击。通过使代码执行栈或缓冲溢出，一名攻击者可尝试得到对服务器的访问。在没有涉及细节的情况下，这样一种攻击调用一个子例程，该例程返回到由攻击者定义的主程序中的一个点。这是几种类似 OS（操作系统）级攻击的一个范例，它利用的是 DNS 服务在其上运行的 OS 弱点。

（6）配置错误。典型情况下，虽然不是恶意的（但多数攻击是从内部源发起的），DNS 服务和/或区域信息的误配置，会导致不正确的解析或服务器行为。

12.1.3　拒绝服务攻击

（1）拒绝服务。像其他网络服务一样，DNS 对于拒绝服务攻击也是脆弱的，这种攻击的特征是，一名攻击者向一台服务器发送数千条报文，希望过载该服务器，导致它的崩溃或对其他查询器是不可用的。该项服务就成为不可用的，因此拒绝对其他查询器的查询做出响应。

（2）分布式拒绝服务。这种类型（指上一条所提到的）攻击的一个变种是，使用多个分布式攻击点，被称作分布式拒绝服务（DDOS）攻击。虽然规模较大，但意图是相同的，可能潜在地影响数台服务器。

（3）反射器攻击。这种形式的攻击，尝试使用 DNS 服务器对一个特定目标发起海量的数据，因此拒绝目标机器的服务。攻击者向一台或多台 DNS 服务器发出许多条查询，在每条 DNS 查询中使用目标机器的 IP 地址作为源 IP 地址。对带有大量数据的记录（例如 NAPTR、EDNS0 和 DNSSEC）查询，会放大这种攻击。每台做出响应的服务器，均以目标为伪造 IP 地址处的“请求者”，以大量数据流淹没这个目标。

12.2　缓解方法

在下表中总结了解决这些弱点的策略。一般而言，您应该对来自厂商的弱点报告做成表格，并当修正和升级可用时，实施修正和升级。对隐藏服务器的部署策略，以及一般而言的部署基于角色的 DNS 服务器，在第 11 章中所讨论的攻击的缓解方面，也是有效的。我们将在下一章详细讨论 DNSSEC。

弱　　点	缓 解 措 施
报文劫持/伪造	DNSSEC 提供了这个弱点的有效缓解措施，方法是提供： 1）源认证：数据源的验证 2）数据完整性验证——接收到的数据与区域文件中发布的数据是相同的 3）存在性的经认证的拒绝——所查找的资源记录不存在
ID 猜测/查询预测	DNSSEC 有效地缓解这个弱点；另外，BIND 9 对 DNS 首部消息 ID 做随机化处理，目的是降低在一条虚假响应中猜测其值的概率。自 2008 年 7 月中旬起，BIND 也在外发查询上对 UDP 端口号做随机化处理，目的是降低这个弱点的风险
名字链/缓存毒化	为了阻止这些弱点，DNSSEC 提供了源认证和数据完整性验证；另外，BIND 对缓存和附加节缓存激活和清除间隔的指令（directive）也是有帮助的；事务 ID 和 UDP 端口随机化也有助于降低这项弱点的风险
解析器配置攻击	通过 DHCP 来配置 DNS 服务器；为了查找误配置或异常，检测或周期地审计各客户端

（续）

弱　点	缓解措施
违法的动态更新	在 allow-update、allow-notify、notify-source 上使用 ACL（访问控制列表）。对于附加的源认证，ACL 也可定义为要求事务签名
违法的区域传递	在 allow-transfer 上与 TSIG（事务签名）一起使用 ACL；对于区域传递，使用 transfer-source IP 地址、端口使用一个非标准端口
服务器攻击/劫持	1）使用隐藏的主服务器来禁止对区域主服务器的检测 2）在主服务器和所有外部 DNS 服务器上禁止（disallow）递归查询 3）使服务器操作系统保持最新 4）限制端口或控制台访问权限 5）实施 chroot
控制通道攻击	在控制语句内使用 ACL，来约束谁能实施 rndc 命令；要求 rndc 密钥
缓冲溢出和 OS 级攻击	保持 OS 最新，限制缓存、附加缓存尺寸，并定义缓存的清除间隔
Named（命名的）服务误配置	使用 checkzone 和 checkconf 设施工具以及带有错误检查的一个 IPAM 系统；如果需要的话，为重新载入保持新的备份
拒绝服务	1）使用速率限制以及各种参数（例如 recursive-clients、max-clients-per-query、transfers-in、transfers-per-ns、缓存和附加缓存尺寸）来限制通信 2）考虑任意播部署
反射器攻击	1）使用 allow-query/allow-recursion ACL 2）如果合适的话，使用视图 3）如果可能的话，在查询上要求使用 TSIG

12.3 非 DNSSEC 安全记录

我们将在下一章讲解 DNSSEC，但以讨论其他面向安全的资源记录类型对 DNSSEC 安全的铺垫一章（即本章）做出小结。

12.3.1 TSIG——事务签名记录

在 RFC 2845[102] 中定义的事务签名（TSIG），使用共享秘密密钥来建立两个 DNS 实体（不管是两台服务器或一台客户端和一台服务器）之间的信任关系。TSIG 提供端点认证和数据完整性检查，并可被用来对动态更新和区域传递签名。TSIG 密钥必须被安全保护，并人工地配置在每个通信端点。

通过在一条 DNS 消息的附加节内包括一个元资源记录类型“TSIG”，就可使用 TSIG 密钥来签名一个事务。类似于用于 EDNS0 的 OPT 资源记录类型，一条元资源记录被用来在一次查询/解析事务过程中传递附加信息，且不包括在一个区域文件本身之中。如此，这些资源记录没有被缓存，对要求签名的消息是动态计算得到的。

TSIG 元资源记录的格式如下：

属主	TTL	类	类型	RData
密钥名	TTL	ANY	TSIG	算法名 签名时间 漂移时间 MAC 尺寸 MAC 原始 ID 错误 其他数据长度 其他数据
k1-k2 ipamww. com.	0	ANY	TSIG	HMAC-MD5. SIG-ALG. REG. INT 23290332 600 32 p19… 5076 0 0

TSIG 元记录内的 RData 字段定义如下。

（1）算法名。以域名格式表示的散列算法名。当前由 IANA 定义的确定算法如下。

1）HMAC-MD5. SIG-ALG. REG. INT（HMAC-MD5）。

2）GSS-TSIG。

3）HMAC-SHA1。

4）HMAC-SHA224。

5）HMAC-SHA256。

6）HMAC-SHA384。

7）HMAC-SHA512。

（2）签名时间。签名时间，是自 1970 年 1 月 1 日 UTC（世界标准时间）时间以来的秒数。

（3）漂移时间（fudge）。在签名时间字段内允许的漂移秒数。

（4）MAC 尺寸。以字节为单位表示的 MAC 长度。

（5）MAC。消息认证码，包含被签名消息的一个散列。

（6）原始 ID。原始消息的 ID 号。如果一条更新被转发，则被转发消息中的消息 ID 可能不同于原始消息的 ID。这样做可使接收者在重构原始消息过程中，利用原始消息 ID 进行签名验证。

（7）错误。对 TSIG 有关的错误编码（见表 9-1）。

1）BADSIG：无效密钥。

2）BADKEY：未知密钥。

3）BADTIME：签名的时间超出漂移范围。

（8）其他数据长度。以字节为单位表示的其他数据节的长度。

（9）其他数据。除非错误 = BADTIME 情况下，为空。在错误 = BADTIME 时，服务器将在这个字段中包括它的当前时间。

基于被签名的消息，构造 TSIG 元资源记录。产生一个摘要，方法是将指定的散列算法应用到消息，并使用这个输出作为 TSIG 资源记录的消息认证码字段。TSIG 元资源记录被添加到 DNS 消息的附加节中。

12.3.2 SIG（0）——涵盖空类型的签名记录

事务签名的另一种形式利用了 SIG 资源记录的一种特殊情形，它是作为 DNSSEC 的初始形式的组成部分而设计的。后来它被 DNSSECbis 中的 RRSIG 资源记录所替换。

不仅如此，一种特殊形式的 SIG 资源记录可独立地用在 DNSSEC 中，用来对更新和区域传递签名。SIG 资源记录的格式如下所示。

SIG（0）表示法是指使用涵盖一个空（即 0）类型字段的 SIG 资源记录。另外，RFC 2931[118] 建议将属主字段设置为根、TTL 为 0 和类为 ANY，如下例所示。

属主	TTL	类	类型	RData
RRSet 域	TTL	ANY	SIG	涵盖类型　算法　标签　原始 TTL　超期　起始时间　密钥标签　签名者　签名
	0	ANY	SIG	0 3 3 86400 20080515133509 20080115133509 30038 ipam-ww. com. q8o1…

12.3.3　KEY——密钥记录

KEY（密钥）记录是由 DNSSEC 的最初定义确定下来的，但后来由 DNSKEY 资源记录替换了。但是，同时在 SIG（0）内也使用 KEY 记录来识别公开密钥，以其对 SIG（0）内的签名解码。KEY 记录与 DNSKEY 记录有相同的格式。

属主	TTL	类	类型	RData
密钥名	TTL	IN	KEY	标志　协议　算法　密钥
K3941. ipamww. com.	86400	IN	KEY	256 3 1 12S9X-weE8F(le…

12.3.4　TKEY——事务密钥记录

虽然 RFC 2845[102] 规范了 TSIG 标准，它利用共享的秘密密钥，但它没有提供一种密钥分发或维护功能。为了支持这项密钥维护功能，人们开发了事务密钥（Transcation Key，TKEY）元资源记录。这个过程开始时，是一个客户端或服务器发送一个签名的[⊖] TKEY 查询，其中包括任意相应的 KEY 记录。来自一台服务器的一条成功响应，将包括一条 TKEY 资源记录，其中包括一个合适的密钥。取决于在 TKEY 记录中指定的模式，现在双方都可确定共享的秘密。例如，如果指定 Diffie-Hellman 模式，则交换 Diffie-Hellman 密钥，双方推算得到共享的秘密，之后用其签名带有 TSIG 的消息。

TKEY 元资源记录的格式如下：

属主	TTL	类	类型	RData
密钥名	TTL	ANY	TKEY	算法名　开始时间　超期时间　模式　错误　密钥尺寸　密钥数据　其他数据长度　其他数据
k1-k2. ipamww. com	0	ANY	TKEY	HMAC-MD5. SIG-ALG. REG. INT 23290332 233006564 2 0 2048 9k)2… 0

⊖ 是的，和在 TSIG 或 SIG（0）中一样签名，所以初始条件下需要一个密钥，但 TKEY 提供删除或更新密钥的一种方法。

TKEY 元记录内的 RData 字段定义如下。

(1) 算法名。以域名格式表示的散列算法名。由 IANA 定义的当前确定的算法如下。

1) HMAC-MD5. SIG-ALG. REG. INT（HMAC-MD5）。

2) GSS-TSIG。

3) HMAC-SHA1。

4) HMAC-SHA224。

5) HMAC-SHA256。

6) HMAC-SHA384。

7) HMAC-SHA512。

(2) 开始时间。密钥有效性开始或起始时间，以自 1970 年 1 月 1 日 UTC 以来的秒数表示。

(3) 超期时间。密钥有效性超期或终止时间，以自 1970 年 1 月 1 日 UTC 以来的秒数表示。

(4) 模式。密钥指派的形式或方案，它可有如下值。

1) 0 = 保留。

2) 1 = 服务器指派。

3) 2 = Diffie-Hellman 交换。

4) 3 = GSS-API 协商。

5) 4 = 解析器指派。

6) 5 = 密钥删除。

7) 6 ~ 65534 = 可用的。

8) 65535 = 保留。

(5) 错误。对 TKEY 有关的错误编码（见表 9-1）。

1) BADSIG：无效密钥。

2) BADKEY：未知密钥。

3) BADTIME：签名时间超出起始/超期范围。

4) BADMODE：指定的模式不支持。

5) BADNAME：无效密钥名。

6) BADALG：指定的算法不支持。

(6) 密钥尺寸。以字节为单位的密钥数据尺寸。

(7) 密钥数据。即密钥。

(8) 其他数据长度。没有使用。

(9) 其他数据。没有使用。

第13章　保障DNS安全（下）：DNSSEC

当我签署一封信或支票时，本质上我在以我的签名说明我同意和授权㊀。当我签署更重要的文档时，例如一个抵押票据，我需要核验我的签名，典型情况下是通过一个公众公证人完成的。公证人验证我的身份，他也核验我的签名，一般是采取这样的方法：将签名与一份驾驶执照或护照签名比对。通过对我的抵押票据盖戳，公证人确认是我签署了该文档，因此我的签名是可被信任的。DNSSEC以一种类似的宽松方式发挥作用。一个解析器或递归服务器，代表一个桩解析器，接收解析数据，还有该数据上的一个签名。只要我信任签名人，那么我就可以使用签名来验证数据。信任的元素要求以被信任密钥的形式，来自签名人的信任信息的某种初始配置，信任密钥被用来验证所接收数据的签名人是值得信任的。如果我不直接信任签名人，则我需要查找签名验证，方法是查找我信任的一个实体，它将“担保”签名人。不但我的抵押公司不信任我，而且他们要求核验我的签名。

DNS安全扩展DNSSEC，最初是在RFC 2535[140]中定义的，后来被修改并重新命名为DNSSECbis，是在RFC 4033～4035[114,141,142]中定义的。这次重新修改是由于原始规范的扩展性问题。虽然仍然一定程度得到改正，DNSSECbis在此之后仍被简单称作DNSSEC，它提供了在DNS内对解析数据源的认证以及验证那个数据完整性的方法。DNSSEC也提供了一种认证DNS数据不存在的方法，这也允许对“没有找到”解析（例如NXDOMAIN）实施签名。因此，DNSSEC支持对报文劫持检测、ID猜测和对解析数据和“没有找到”解析的缓存毒化攻击。DNSSEC通过使用非对称公开密钥密码学技术来提供这些服务，实施数据原始认证和端到端数据完整性验证。

13.1　数字签名

我们在第10章DKIM的上下文中，介绍了数字签名产生和验证的概念和过程，但在这里出于方便查看的目的，我们将简短地回顾一下。数字签名使一个给定数据集的源发者，使用一个私有密钥对数据签名，从而使接收数据和签名，以及用于解密签名的一个对应公开密钥的那些接收方，可实施数据源发和完整性验证。DNSSEC使用一个非对称密钥对（私有密钥/公开密钥）模型。在这样一个模型中，采用一个私有密钥签名的数据可通过采用对应的公开密钥对数据解密的方法，来加以验证。私有密钥和公开密钥形成一对密钥。从概念上来说，私有/公开密钥对为公开密钥的持有者提供了验证数据的一种方法，该数据是使用相应的私有密钥签名的。这提供了数据确

㊀　这个基本介绍来自参考文献［11］的第9章，其中的高层描述在本章得到了比较详细的扩展讨论。

实是由私有密钥持有者签名的认证。数字签名也支持这样的验证，即所接收的数据匹配发布的数据，且在中转过程没有被篡改。

见图13-1，数据源发者，如图左侧所示，产生一个私有密钥/公开密钥对，并利用私有密钥对数据签名。签名数据的第一步是产生该数据的一个散列，有时也被称作一个摘要。散列函数是一种单向函数[⊖]，它将数据扰乱成一个固定长度的字符串，以便进行比较简单的操作运算，并代表数据的一个“指纹”。这意味着，要想另一个数据输入可产生相同的散列值，是非常不可能的情况。因此，散列通常被用作检验和，但不提供任何原始认证（知道散列算法的任何人均可简单地对任意数据执行散列运算）。常见的散列算法包括HMAC-MD5、RSA-SHA-1和RSA-SHA-256。使用私有密钥对散列值进行加密，产生签名。加密算法以散列值和私有密钥为输入，产生签名。

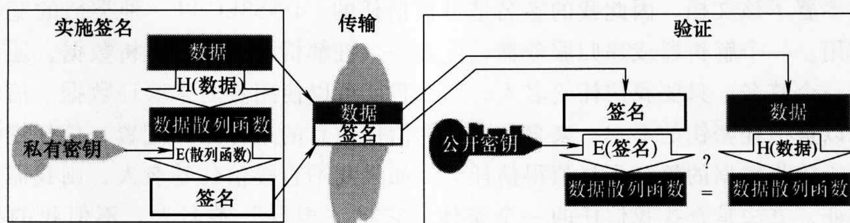


图13-1 数字签名产生和验证过程[11]

数据及其相关联的签名被传输到接收者。注意数据本身没有被加密，仅简单地被签名。接收者必须可访问公开密钥，它对应于用来签名数据的私有密钥。在一些情形中，使用一个安全的（被信任）公开密钥分发系统（例如公开密钥基础设施(PKI)），使公开密钥是可用的。在DNSSEC的情形中，在DNS内发布公开密钥，除此外，还有解析信息和对应的签名。

就和数据源发者一样，接收者计算所接收数据的一个散列值。通过使用源发者的公开密钥，接收者将加密算法施用到接收到的签名。这种运算是签名产生过程的反过程，并产生原始数据散列作为它的输出。这种解密的输出，即原始数据散列值，和接收者对数据计算出的散列进行比较。如果相匹配，则数据没有被修改，是私有密钥持有者对数据进行的签名。如果私有密钥持有者可被信任，则认为数据是经过验证的。

13.2 DNSSEC综述

DNSSEC利用这种非对称密钥对的密码学算法，提供数据源发认证和端到端数据完整性确认。伪造或沿路到目的地篡改数据的任何尝试，都将被接收者检测到，这种接收者是解析器，或比较典型情况下，是代表其自身的递归/缓存DNS服务器。这种功能特征，使DNSSEC针对中间人和缓存毒化攻击而言，成为一项有效的缓解策略。

原始DNSSECbis规范没有考虑一种安全的密钥分发系统，所以一个或多个被信

⊖ 一个单向函数意味着，不能从散列值唯一地得到原始数据。即人们可将一个算法施用，产生散列值，但不存在反向算法，可在散列值上实施，得到原始数据。

任密钥必须人工地配置在解析器或递归名字服务器上㊀。但是，一个后来的规范，RFC 5011[141]定义了方便这个过程的一种方法，方法是依据一个手工配置的初始密钥，认证新的和撤销的信任密钥。这个初始密钥用作“初始条件”，用在随新密钥、撤销密钥和删除密钥的时间向前推移而推进的过程。稍后我们将讨论这个自动化的信任密钥更新过程。无论是人工配置的或是自动更新的，每个信任密钥均识别对应于一个给定信任区域的公开密钥，该区域由区域管理员授权的。

这类似于银行公证人，银行信任他来验证我的身份。毕竟，任何冒名顶替者均可以用一个私有密钥签名无效（非法）的区域数据，并发布对应的数据、签名和公开密钥。因此，递归服务器必须预先配置一个密钥或一组密钥，它们是被信任的，对应于被信任的签名区域。信任区域管理员所用的当前公开密钥，必须以带外方式传输到解析器管理员，或使用 DNS 之外的一种机制做到这点。采用刚才提到的自动密钥更新过程，针对每个信任区域，必须配置一个初始密钥；但是，使用 DNS 协议可实施不断进行的密钥更新。

一个给定的信任区域可认证一个子区域的公开密钥，将信任模型从仅仅是信任区域扩展到信任区域及其所认证的子区域。类似地，这些子区域可认证它们的子区域等，这样就从信任区域到所有签名的委派区域形成了一个信任链。随着现在因特网根区域被签名、大型 TLD 被签名或很快会被签名，则信任链将从根信任锚点发出到 TLD，沿域树向下到较低层次的被签名区域。

信任密钥的配置，要求一名区域管理员得到他的/她的公开密钥，产生一个信任关系。采用一个被签名的根和 TLD（多个 TLD），这种做法简化了信任模型，这要求信任根区域和根区域信任锚点的配置。作为替代（实际上作为后者的祖先之一，即先于后者）使用根区域密钥，ISC 也构造了一个信任的密钥注册机制（dlv. isc. org），作为注册域的被信任密钥的一个库，这种做法支持作为一个“父区域代理”过程中的“旁查”（lookaside）验证，这降低了对每个域管理员形成个体关系的外部需求影响。

虽然 DNSSEC 规范没有明确要求，但从运营经验得到这样的建议，即每个区域使用两个密钥，一个区域签名密钥（ZSK）和一个密钥签名密钥（KSK）㊁。如我们后面将看到的，这加速了复杂密钥轮转（rollover）进程，同时在密钥长度安全性和复杂性，与依据需求灵活改变区域签名秘密之间做出了折中。此时，可以说，ZSK 被用于区域内部签名数据，KSK 是一个较长期的密钥，用其签名 ZSK。ZSK 和 KSK 均由一个公开密钥和私有密钥对组成。私有密钥被用来签名区域信息，必须是得到安全保障的，理想情况下，位于一台安全的服务器或主机之上。对应的公开密钥，是以 DNS-KEY 资源记录的形式在区域文件内发布的。

被信任区域的公开 KSK 是配置在每台递归服务器上的信任密钥㊂，它应该匹配发

㊀ 从实用主义角度而言，术语“解析器”在 DNSSEC 上下文中是指递归服务器的解析器功能，它也解析被查询的信息，验证签名。考虑第 11 章中我们的实施例，因特网缓存服务器将执行这项签名验证功能。

㊁ 在 RFC 4641[167]中讨论了这个建议的动机以及其他 DNSSEC 运营实务的讨论。

㊂ 一个信任密钥与一个信任锚点是同义的，它也被称作 DNS 域树中的安全入口点（SEP）。

布于相应区域文件中的对应 KSK DNSKEY 资源记录。使用区域的 ZSK 来验证被解析数据的签名，被信任的 KSK 对 ZSK 签名，并由此可对 ZSK 的签名进行验证。

如果不信任这个 KSK，则尝试检查父区域是否被签名，或旁查验证是否配置。如果被签名，则这个父区域（或旁查注册机制）将其到子节点的委派实施签名，方法是以父区域中一条委派签名者（DS）记录（或 DNSSEC 旁查验证，DLV 记录）的形式，对子区域的公开 KSK 签名。接下来这个委派是以父区域的 ZSK 签名的，而 ZSK 自己是以父区域的 KSK 签名的。同样，如果这些签名是有效的，且 KSK 匹配一个信任密钥，则解析是完备的和安全的。否则，该过程继续到父区域的父区域等。

验证过程沿信任链向上，直到遇到一个匹配的信任密钥，如果找到这样的一个密钥，则认为解析数据是经过核验的（validated）。否则，它将将被认为是不安全的。

13.3 配置 DNSSEC

实现 DNSSEC 的过程涉及生成私有/公开密钥对、将公开密钥信息添加到要被签名的区域文件、以对应的私有密钥对区域签名和将公开 KSK 信息分发到父区域管理员或解析器管理员，这些人信任您和您的区域信息。图 13-2 形象地说明了基本过程。

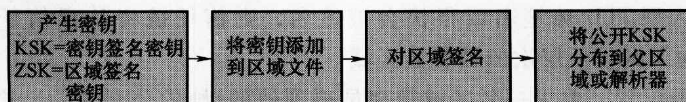


图 13-2 基本的 DNSSEC 实现步骤[11]

现在让我们形象地说明这个基本过程，检查研究一下实现 DNSSEC 的机制。我们使用人工的和自动的㊀BIND 方法和工具[144]来形象地说明该过程，如今这些方法和工具支持 DNSSECbis。微软在其 Windows Server 2008 R2 发行版中支持 DNSSECbis。在比较详细地回顾这些步骤之后，通过签名 ipamworldwide. com 区域文件，我们将展示说明 DNSSEC 的实现。

13.3.1 产生密钥

我们的第一步是产生密钥，将用其对我们的区域信息签名。BIND 发行时带有 dnssec-keygen 设施工具，它提供了产生一个私有/公开密钥对的一个简单命令行。它甚至生成 DNSKEY 记录。为了生成我们的 ipamworldwide. com 区域的一个 ZSK 密钥对，我们使用 dnssec-keygen 命令：

```
dnssec-keygen-a RSA-SHA-1-b 1024-n ZONE-c IN-e ipamworldwide. com
```

这个设施工具不仅用来生成 DNSSEC 密钥，而且可用来生成 TSIG 密钥和 KEY 记录。除非明确指定，所有参数都是可选的，在 dnssec-keygen 设施工具内的参数包括如下内容。

㊀ BIND 9.7.0 引入了几项新的密钥和签名管理功能特征，以其使这些步骤自动化，就和我们将要描述的情形一样。

（1）-a 算法：（必需的）其中 DNSSEC 密钥的算法可以是如下内容。

1）RSA-SHA-1。

2）RSA-SHA-256。

3）RSA-SHA-512。

4）NSEC3-RSA-SHA-1（带有一个符号（signal）的 RSA-SHA-1 算法，指明以这个密钥签名的区域可使用 NSEC3）。

5）DSA（数字签名算法）。

6）NSEC3DSA（带有一个符号（signal）的 DSA 算法，指明以这个密钥签名的区域可使用 NSEC3）。

7）RSA-MD5。

（2）-b 密钥尺寸：（必需的）指定了密钥中的比特数。每种算法的有效密钥尺寸如下。

1）RSA-SHA 密钥。512～2048bit。

2）DSA 密钥。512～1024bit，可被 64 整除。

（3）-n 名字类型：（必需的）识别密钥属主的类型。有效值包括 ZONE（对于 DNSKEY 是默认的）、HOST、ENTITY、USER 或 OTHER。

（4）-3：使用支持 NSEC3 的密钥产生算法（如果指明没有-a 参数，则使用 NSEC3-RSA-1）。RSA-SHA-256 和 RSA-SHA-512 也是支持 NSEC3 的。

（5）-A 日期/偏移：设置密钥的激活日期。当表示为 YYYYMMDD 或 YYYYMM-DDHHMMSS 格式（或没有设置时为 none（无））时，日期/偏移字段是一个绝对日期/时间；或当以这些日期格式之一带有一个“+”或“-”前缀时，这时日期/偏移字段是一个距离当前时间的一个偏移。当没有设置时，且-G 也没有设置时，默认是“now”（现在时间）。

（6）-C：产生私有密钥的选项，前提条件是没有有关生成、发布和/或激活日期的任何元数据，这可能与较老的 BIND 版本不兼容。

（7）-c 类：包含该密钥的 DNS 资源记录的类。

（8）-D 日期/偏移：定义了这个密钥从这个区域被删除时的日期或距离当前时间的偏移。当表示为 YYYYMMDD 或 YYYYMMDDHHMMSS 格式（或没有设置时为 none（无））时，日期/偏移字段是一个绝对日期/时间；或当以这些日期格式之一带有一个“+”或“-”前缀时，这时日期/偏移字段是一个距离当前时间的一个偏移。在指定时间，该密钥将从区域被清除，但它将保留在密钥库中。

（9）-e：当使用 RSA-MD5 或 RSA-SHA-1 算法时，使用一个大指数的命令选项。

（10）-E 引擎：使用密码学硬件（OpenSSL 引擎）用于随机数生成和当支持密钥生成时的命令选项。当采用 PKCS#11 支持进行编译时，默认为 pkcs11，否则为 none（无）。

（11）-f 标志：在 DNSKEY（或 KEY）资源记录中设置标志字段；目前，标志 = KSK 被用来产生一个 KSK（在 DNSKEY 记录中设置 SEP 比特）；标志 = REVOKE 将为这个密钥设置撤销标志。

（12）-g 生成器：为 DH 算法指定一个密钥生成器值。

（13）-G：生成一个密钥，它不用于发布或用来签名。

（14）-h：打印这条命令的一个帮助（help）摘要。

（15）-I 日期/偏移：设置密钥退役（retired）时的日期/偏移。当表示为 YYYYMMDD 或 YYYYMMDDHHMMSS 格式（或没有设置时为 none（无））时，日期/偏移字段是一个绝对日期/时间；或当以这些日期格式之一带有一个“+”或“-”前缀时，这时日期/偏移字段是一个距离当前时间的一个偏移。当退役时，密钥保留在区域中，但不再用来对区域签名。

（16）-k：指明要产生的是一条 KEY 记录，而不是 DNSKEY 记录；为了使用-T 选项，这个选项被废弃了。

（17）-K 目录：定义了密钥文件将被放置于其中的目录。

（18）-p 协议：设置资源记录中的协议字段。对于 DNSSEC，使用默认值 3。

（19）-P 日期/偏移：设置密钥在区域文件中发布（但并不用来对区域签名）的日期/偏移。当表示为 YYYYMMDD 或 YYYYMMDDHHMMSS 格式（或没有设置时为 none（无））时，日期/偏移字段是一个绝对日期/时间；或当以这些日期格式之一带有一个“+”或“-”前缀时，这时日期/偏移字段是一个距离当前时间的一个偏移。

（20）-q：静默模式，这种模式抑制输出，包括指明进度的信息。

（21）-r 随机源：指明一个源或随机数据，例如一个文件或字符设备（例如键盘）。

（22）-R 日期/偏移：定义密钥被撤销时的日期。当表示为 YYYYMMDD 或 YYYYMMDDHHMMSS 格式（或没有设置时为 none（无））时，日期/偏移字段是一个绝对日期/时间；或当以这些日期格式之一带有一个“+”或“-”前缀时，这时日期/偏移字段是一个距离当前时间的一个偏移。当撤销时，在相应 DNSKEY 资源记录中设置“撤销”比特，但密钥将保留在区域中并被用来对区域签名。

（23）-s 强度：指定密钥的强度值，但不与 DNSSEC 相关。

（24）-t 类型：指明使用密钥来认证数据（AUTH）和/或加密数据（CONF）。AUTHCONF 支持这两项功能，但 NOAUTH、NOCONF 和 NOAUTHCONF 不支持对应的功能。

（25）-T rrtype：以资源记录格式指明公开密钥产生所用的 RRType。rrtype 的有效值包括 DNSKEY（默认）或 KEY。

（26）-v 等级：设置 debug（调试）等级。

（27）密钥名：（必需的）密钥的名字，一般而言是区域名，它可用作 DNSKEY 记录的属主字段。

五个基于日期/偏移的选项在 BIND 9.7.0 中是新的，并为所生成的密钥提供定时的元数据，因此提供了在区域签名过程中被使用的时间信息。因此，在密钥的整个生命周期中，密钥可被分阶段并轮转（roll）使用，生命周期由在区域中发布的、用于签名区域信息、撤销、退役和删除等阶段组成。在一个密钥的生命周期中，对这些选

项及其用途进行简要描述，如下。

（1）-P。定义所生成密钥在区域文件中发布的时间，但在签名中不使用该密钥。

（2）-A。定义激活时间，这是所产生密钥被用于签名区域数据的时间。如果这是一个 KSK，且 update-check-ksk 选项设置为是，则这个密钥将仅签名 DNSKEY RRSet。否则，它将被用来签名所有的区域 RRSet，这就允许单一区域密钥实施的情形出现。

（3）-R。定义这个密钥被撤销的时间。这就定义了撤销标志将被设置在相应的 DNSKEY 记录中的日期。这个密钥（带有设置好的撤销比特）将仍然被用来签名区域文件，但解析器将被通知"这个密钥被撤销了"。

（4）-I。定义该密钥的退役时间，在此时间之后，这个密钥将不被用来签名区域数据，但它将保留在区域文件中。

（5）-D。定义密钥将从区域文件中删除的时间。

这些定时（timing）选项使您能够在密钥生成时就定义整个密钥生命周期！

另一种密钥文件生成设施工具，首次被包括在 BIND 9.6.0 发行版之中，它允许使用 PKCS#11[⊖] API 与一个密码学令牌生成硬件设备接口。dnssec-keyfromlabel 设施工具从密码学硬件设备得到密钥，并生成公开密钥和私有密钥文件。这个工具有如下参数，和 dnssec-keygen 的格式完全相同，增加了一个新的必备参数，指明密钥标签。

（1）-a 算法：（必备）与 dnssec-keygen 的值相同。

（2）-3：与 dnssec-keygen 的含义相同。

（3）-c 类：与 dnssec-keygen 的值相同。

（4）-C-：与 dnssec-keygen 的含义相同。

（5）-E 引擎：与 dnssec-keygen 的值相同。

（6）-f 标志：与 dnssec-keygen 的值相同。

（7）-G：与 dnssec-keygen 的含义相同。

（8）-C-：与 dnssec-keygen 的含义相同。

（9）-h：与 dnssec-keygen 的含义相同。

（10）-k：与 dnssec-keygen 的含义相同。

（11）-K 目录：与 dnssec-keygen 的含义相同。

（12）-l 标签：（必备）在 PKCS#11 设备上的密钥标签。

（13）-n 名字类型：与 dnssec-keygen 的值相同。

（14）-p 协议：与 dnssec-keygen 的值相同。

（15）-t 类型：与 dnssec-keygen 的值相同。

（16）-v 等级：与 dnssec-keygen 的值相同。

（17）-y：即使密钥 ID 与一个现有密钥冲突，也允许产生 DNSSEC 密钥文件。

（18）密钥名：（必备）。

也支持上面讨论的元数据定时选项。dnssec-keygen 和 dnssec-keyfromlabel 都返回

⊖ PKCS#11 是公开密钥密码学标准家族的一员，该标准由 RSA 实验室发布。

密钥名。在我们的例子中，结果是

Kipamworldwide. com. +005 +14522

密钥名的格式遵循这个惯例

（1） K（用于密钥）。

（2） 密钥名（ipamworldwide. com）。

（3） 密钥产生算法（在这种情形中是 005 = RSA-SHA-1）。

（4） 密钥标签（tag）或密钥的身份（14522）。

密钥标签提供了索引密钥的一种简便方法，我们稍后将看到。由 dnssec-keygen 或 dnssec-keyfromlabel 产生两个文件，一个带有扩展名 . private，指明是私有密钥，另一个带有扩展名 . key，它以一条 DNSKEY 记录的形式包含公开密钥。在我们的例子中，这两个密钥文件如下命名。

Kipamworldwide. com. +005 +14522. private

Kipamworldwide. com. +005 +14522. key

Kipamworldwide. com. t005t14522. private 文件包含私有密钥细节，包括格式、算法、模块、指数、素数和系数值，如下 dnssec-keygen 命令的输出（为了改进可读性，插入了空行）所示。

```
Private-key-format: v1.2
Algorithm: 5 (RSASHA1)
Modulus:
x6QAwJiz6hHa/eUI2pGz6rvwEYpJdi1TJH8Uj4lDPTmzseCOgFEqB3/dZB0Q
  5LEs1ZetAJJEk4F + WccRKwqnIcGkvIKfTC8hn + gbiBAnadQRFLxNMBs6KB0e +
  yqiNK60sbrn22F8AYRiG3n2rTQndVtkaZep9jbcCqfu/DagB10 =
PublicExponent: AQAAAAE =
PrivateExponent:
CWheqbbkIx3kRIa7NyDbdwZYGA83uBtdfnBTu8QyV8/h419T3fyWrWfKo4wi
  Vys9ql0Xmumwy/hSLmZJJrzxS6SVwaM/iEunsyyiHedeVKiMeYVlOlvJ3 +
  OweKy/59y3drJS + qAm + cbtrhWZheXtzgR78wp2IK + 4kHAhZTCYGAE =
Prime1:
8YuU4sicmKmu5Cz4IUjvE2kQit5pJPV3yUK04nPz9P0MJFKyCIAdsw2A5HoRn3 + +
  I5BtDjeQxkD0aFGA4S0fKXQ = =
Prime2:
05ZzyiaiZK1JqQMCgT977NZkEuKgXI4seTUL1Wu7Z/FRs/7xHE4oSJrx7siwLOx
  WJKcc4Fo + 4erVRHioiOadhAQ = =
Exponent1:
Hpy1z37UsfdONCV7Kd/8xu07PslhtbX7EFVGRno/dOrWNp5p64hVhF5tbnNBVz
  ZHRQ + 5IZzwMfQ3A3 + GjY8QQQ = =
Exponent2:
jfw + s9zt8uVMwubwowwxOsjX32GO3VrSPk68 + CisiAVxYS8EdTOqvpYps6Vz +
```

rJNnnk45urnlqDbWCx2tugyAQ = =

Coefficient：

uUC/aKgEvOQymCmMukC4ExTm/7ly2w31V/NMOF2GzC7fc1gYvDZEOX6YNnz5e8

PRD2bQXCTgsMorRs7PJYI2Cg = =

Kipamworldwide. com. t005t14522. key 文件理解起来要比较容易点，它包含我们的DNSKEY资源记录

ipamworldwide. com. IN DNSKEY 256 3 5

BQEAAAABx6QAwJiz6hHa/eUI2pGz6rvwEYpJdi1TJH8Uj4lDPTmzseCO

gFEqB3/dZB0Q5LEs1ZetAJJEk4F + WccRKwqnIcGkvIKfTC8hn + gbiBAn

adQRFLxNMBs6KB0e + yqiNK60sbrn22F8AYRiG3n2rTQndVtkaZep9jbc

Cqfu/DagB10 =

DNSKEY资源记录的解释或格式描述如下。

属主	TTL	类	类型	RData
区域名	TTL	IN	DNSKEY	标志 协议 算法 密钥
ipamworldwide. com.	86400	IN	DNSKEY	256 3 5 BQEAAA…

属主字段定义区域名。RData由如下子字段组成。

（1）标志。指明这个密钥是一个区域密钥（值=256）。当前为标志字段定义的值如下。使用如下的十进制值，我们看到一个ZSK将有一个标志值256，而一个KSK将有一个奇数值，像257。

1）第7bit。ZSK（十进制=256）。

2）第8bit。撤销签名（十进制=128）。

3）第15bit。KSK或安全入口点（SEP）（十进制=1）。

4）其他bit。未指派。

（2）协议。必须有一个值“3”，指明是DNSSEC（这是当前定义的唯一值）。

（3）算法。定义密钥生成中使用的算法。当前支持的算法编码如下。

1）值=1。RSA-MD5，依据RFC4034，不建议使用该值。

2）值=2。Diffie-Hellman。

3）值=3。DSA-SHA-1。

4）值=4。保留用于椭圆曲线算法。

5）值=5。RSA-SHA-1，依据RFC 4034，这是必备的算法。

6）值=6。DSA-NSEC3-SHA1。

7）值=7。RSASHA1-NSEC3-SHA1。

8）值=8。RSA-SHA-256。

9）值=10。RSA-SHA-512。

10）值=12。GOST R34. 10-2001。

11）值=252。间接的。

12）值253-254。私有的。

13）值＝0，255。保留的。

14）其他值。未指派的。

（4）密钥。公开密钥（ZSK或KSK）。

现在我们重复dnssec-keygen命令，这时使用-f KSK参数，还有一个较长的密钥尺寸，来产生我们的KSK对。

dnssec-keygen-a RSASHA1-b 2048-n ZONE-c IN-e-f KSK ipamworldwide. com.

命令行对这条命令的响应是密钥对名，Kipamworldwide. com. t005t06082。

产生所得DNSKEY记录

ipamworldwide. com. IN DNSKEY 257 3 5

AwEAAdSAwGoUBhtjpE8GLGN4ryt8yEq71DqdE + ij3boe9lmvpM02YZ1/
AQxoHbyA7NqRr + 8dsTM8OrF2yFRbcP1y0/9q37T0PqxL5HjAZ8HrDoW9
R/pC3XyRe9pMzRNr4as + c/xEISfhxzvR84CndF5XvFeh3H0kVDeTb + 7Q
RrG7hnph4P8w4SMg76tBvxHLFmj3OdP8vIUpRAnexEAdclamj1ZSPjLc
dICzpDvQB/LLsYxx8wx2h0vTvhxZklqmy1dPBtIZu2A551VIrU0xgCJx
DjJGCgBbrp1C01tYSdqlA1I2HCL8eV7io/CxnCuSThPlXaPLySojJpXU
gDomWgVYeo0 =

注意由于设置了SEP标志，所以对于KSK，标志字段值为257，对于ZSK，是256。我们将以其keyid＝06082（来自所生成的密钥名）来指称KSK，以其keyid＝14522来指称ZSK。

13.3.2　将密钥添加到区域文件

在我们对区域签名之前，我们需要将我们的两条DNSKEY资源记录包括在区域文件内。因为密钥文件包含我们的DNSKEY资源记录，所以您可从文件中剪切并粘贴到区域文件，或简单地对每个文件使用一个$ INCLUDE语句。

$ INCLUDE Kipamworldwide. com. +005 +14522. key

$ INCLUDE Kipamworldwide. com. +005 +06082. key

也不要忘了增加您的序列号。在以dnssec-signzone设施工具对区域签名之前，首先运行namedcheckzone，这是一个好的想法。

13.3.3　对区域签名

区域签名过程利用了另一个BIND设施工具dnssec-signzone，它实施许多个功能来对区域签名。首先，它规范化地对区域内的资源记录排序。从本质上来说，这是对区域内的资源记录按字母顺序排列。这有利于签名应用以常见属主名、类和类型将资源记录归组到资源记录集合（RRSet）。对资源记录进行规范化排序的另一个原因是识别区域文件内RRSet间的间隔（gap）和带有下一个安全资源记录的总体（population），安全资源记录提供在一个区域内一条给定资源记录的经过认证的存在性拒绝信息。一条NSEC3PARAM资源记录必须存在于区域文件中，以便在区域签名过程中生成NSEC3记录。

在规范化排序和插入NSEC［3］记录之后，dnssec-signzone对区域文件内的RR-Set签名，包括DNSKEY RRSet（以便被$ INCLUDE在我们的例子中）和NSEC［3］RRSet。被签名的区域文件包含原始的RRSet，是规范化排序的，并使用资源记录签名（RRSIG）记录进行了签名。文件也包括一条NSEC［3］记录以及在文件内每个RRSet所对应的RRSIG记录。在区域文件内没有被签名的唯一记录是子区域的NS记录。子区域（而不是父区域）是这个信息的权威；因此，父区域没有认证这些消息的准确性。

幸运的是，dnssec-signzone设施工具实施所有这些步骤，自动地进行规范化排序、NSEC［3］插入以及RRSIG构造和插入，以便产生一个被签名的区域。这里是dnssec-signzone命令，我们将用之对ipamworldwide. com. 区域签名。

dnssec-signzone-k Kipamworldwide. com. +005 +06082-l dlv-registry.

net-g-o ipamworldwide. com. -t db. ipamworldwide. com Kipamworldwide.

com. +005 +14522. key

dnssec-signzone设施工具的参数包括如下内容。

（1）-3 salt（精选值）：当对这个区域签名时，使用指定的salt值，生成一个NSEC3链。salt是以十六进制格式指定的，一个短线（-3-）指明，当生成NSEC3链时，不应使用salt。

（2）-a：验证所有生成的签名。

（3）-A：当生成一个NSEC3链时，在所有NSEC3记录上设置OPTOUT标志，且不要对未签名的子区域（不安全的委派）生成NSEC3记录。

（4）-c类：DNS区域的类。

（5）-C：与dnssec-signzone的较陈旧版本的兼容模式；在签名zonename区域时，除了dssec-zonename之外，生成-zonename密钥集合。

（6）-d目录：为了对区域签名，查找dsset或keyset文件的指定目录。

（7）-e end_ time：指定所生成的资源记录集合签名记录过期时的日期和时间。End_ time可使用 $+N$ 以相对于当前时间的方式指定，其中 N 是距离当前时间的秒数，或使用格式YYYYMMDDHHMMSS的协调统一时间（Coordinated Universal Time，UTC）的绝对时间表示。当忽略这个参数时，默认end_ time是距离start_ time的30天时间（见-s）。

（8）-E引擎：为区域签名，使用密码学硬件（OpenSSL引擎）的命令选项，使用的是来自一个安全密钥存储（当支持时）的密钥。当采用PKCS#11支持进行编译时，默认的引擎是pkcs11，否则为none（无）。

（9）-f文件：指定被签名区域的文件名。如果忽略，则默认是附加有签名的当前区域文件名。

（10）-g：指明应该产生委派签名者资源记录，由该记录认证所签名的子区域；得到的ds-set密钥集合，被提供给父区域的管理员，包括在用于签名的父区域中。

（11）-h：打印这条命令的help摘要描述。

（12）-H iterations（重复次数）：当生成一条NSEC3链（当指派-3选项时）时，

使用 iterations 次的重复数。

（13） -i 间隔：当对一个区域重新签名（传递一个以前签名的区域作为输入）时，间隔指定任何签名记录超期前距离当前时间的时间间隔，此时将重新产生间隔。因此，如果签名（RRSIG）记录被设置为在 5 天内过期，且该区域被重新指派一个 6 天的间隔，则将重新产生签名记录；否则，将保留当前的签名。

（14） -I 输入格式：定义要签名的区域文件的输入格式，文本格式（默认的）或原始的（raw）。将这个选项设置为原始的，这有助于对原始区域数据的签名，该数据包括动态更新，因此为静态区域增加不了多少价值。

（15） -j 抖动（jitter）：支持指定一个窗口，该窗口被用来随机化 RRSIG 签名超期时间，目的是降低几个同时超期的影响，当被签名区域的时间过了、需要重新指派时，每个超期都需要重新产生签名。

（16） -k 密钥：指派的密钥是一个 KSK；可提供多个-k 参数。

（17） -K 目录：定义密钥文件所处的目录。

（18） -l 域：生成一个 DLV 密钥集合文件；可将这个密钥集合注册到 DLV 注册结构，以便验证这个区域的“委派”。

（19） -n threads（线程数）：指定当实施这项操作时，要使用的 CPU 线程数。

（20） -N serial-format（序列号格式）：指定被签名区域的 SOA 记录序列号的格式，为如下之一。

1）保持原样（keep）：不修改区域文件输入的序列号。

2）增加（increment）：依据 RFC 1982 序列号算法，增加序列号。

3）unixtime：将序列号设置为自计时开始以来（自 1970 年 1 月 1 日子夜 UTC 以来的时间，不计算闰秒）的秒数。

（21） -o 源始点（origin）：指定被签名区域的区域原始点。

（22） -O 输出格式（output-format）：指派被签名区域的输出格式为文本（默认的）或原始的。

（23） -p：当对区域签名时，使用伪随机数据，相比于依据-r 参数而使用真实随机数据的方法，这种方法要快速但不太安全。

（24） -P：禁止默认的签名后（postsigning）验证测试，包括对每个正在使用的算法验证存在一个有效的未撤销的 KSK、验证所有被撤销的 KSK 都是自签名的和针对每个算法区域中的所有记录都是签过名的。

（25） -r 随机源：指明一个随机数据源，例如一个文件或字符设备（例如键盘）。

（26） -s start_ time：指定资源记录集合签名记录（RRSIG）成为有效的日期和时间。start_ time 可使用 +N 以相对于当前时间的方式指定，其中 N 是距离当前时间的秒数，或使用格式 YYYYMMDDHHMMSS 的协调统一时间（Coordinated Universal Time，UTC）的绝对时间表示。当忽略这个参数时，默认 start_ time 是距离当前时间的 1h，目的是运行时钟偏差。

（27） -S：“灵巧签名”利用密钥元数据，使用 dnssec-keygen 的定时选项加以配置；搜索密钥库查找与要签名的区域相匹配的密钥，将它们包括在与相应元数据和定

时关系一致的区域文件内，之后对区域签名。满足如下条件的密钥被用来对区域签名，即当前日期已经过了激活日期或撤销日期，但在退役或删除之前（或如果不存在元数据）；满足如下条件的密钥可被发布但不被用来对区域签名，即当前日期过了发布日期但在其他日期之前。

（28）-t：在签名过程完成时，打印统计信息。

（29）-T ttl：定义与 DNSKEY 记录一起使用的 TTL 值（如果在区域中任何未废弃的 DNSKEY 记录上没有指定 TTL 的话），作为灵巧签名的组成部分（见-S），这些记录是从密钥库输入到区域文件中的。

（30）-u：更新区域内的 NSEC［3］链；同样也支持从一个 NSEC 链式区域到一个 NSEC3 链式区域的切换，反之亦然，但这取决于区域文件中是否存在 NSEC3PARAM 记录。

（31）-v level：设置调试（debug）等级。

（32）-x：以 KSK 对区域的 DNSKEY RRSet 签名，但并不带 ZSK。

（33）-z：当确定要对什么签名时，忽略 KSK 标志（SEP 标志比特）；即使有 KSK［和 ZSK］对区域 RRSet 签名也要忽略。

（34）zone_ file：要签名的区域文件名。

（35）密钥：用来对区域数据签名的密钥。

Dnssec-signzone 设施工具的输出是被签名的区域，它使用相同名字作为原始的未签名区域，串接一个“.signed”后缀。看看我们的例子，您可如下看到 db. ipamworldwide. com. signed 文件要远远大于我们的原始区域文件。考虑在签名前我们的初始 db. ipamworldwide. com 文件

```
$ TTL86400
ipamworldwide. com. 1D IN SOA extdns1. ipamworldwide. com.
    dnsadmin. ipamworldwide. com. (
                204 ; serial
                3H ; refresh
                15 ; retry
                1w ; expire
                3h ; minimum
                )
ipamworldwide. com. 86400 IN NS extdns1. ipamworldwide. com.
                86400 IN NS extdns2. ipamworldwide. com.
                86400 IN NS extdns3. ipamworldwide. com.
extdns1. ipamworldwide. com. 86400 IN A 192. 0. 2. 34
                86400 IN AAAA 2001:db8:4af0:2010::a
extdns2. ipamworldwide. com. 86400 IN A 192. 0. 2. 42
                86400 IN AAAA 2001:db8:4af0:2011::11
extdns3. ipamworldwide. com. 86400 IN A 192. 0. 2. 50
```

```
                86400 IN AAAA 2001:db8:4af0:2006::9
eng.ipamworldwide.com. 1w IN NS ns1.eng.ipamworldwide.com.
                1w IN NS ns2.eng.ipamworldwide.com.
ns1.eng.ipamworldwide.com. 1w IN AAAA 2001:db8:4af0:2007::7
ns1.eng.ipamworldwide.com. 1w IN AAAA 2001:db8:4af0:2009::12
$ORIGIN ipamworldwide.com.
                1D IN MX 10 smtp1.ipamworldwide.com.
                1D IN MX 20 smtp2.ipamworldwide.com.
www             1D IN A 192.0.2.37
                1D IN AAAA 2001:db8:4af0:2010::25
                1D IN A 192.0.2.53
                1D IN AAAA 2001:db8:4af0:2006::5
w3              1D IN CNAME www.ipamworldwide.com.
smtp1           1D IN A 192.0.2.36
                1D IN AAAA 2001:db8:4af0:2010::1b
smtp2           1D IN A 192.0.2.45
                1D IN AAAA 2001:db8:4af0:2011::2b
ftp-support     1D IN A 192.0.2.44
                1D IN AAAA 2001:db8:4af0:2011::2c
$INCLUDE Kipamworldwide.com.+005+14522.key
$INCLUDE Kipamworldwide.com.+005+06082.key
```

将上述内容与如下签名版本相比对：

```
ipamworldwide.com. 86400 IN SOA extdns1.ipamworldwide.com.
dnsadmin.ipamworldwide.com. (
                        204 ; serial
                        10800 ; refresh (3 hours)
                        15 ; retry (15 seconds)
                        604800 ; expire (1 week)
                        10800 ; minimum (3 hours)
                        )
            86400 RRSIG SOA 5 2 86400 20100305135354 (
                        20100203135354 14522 ipamworldwide.com.
                        OQS+AaE57+ffRfz+SaMHOJI6b4l2bNnsSDIK
                        mIIMdmXOw8cylCMieaUBz8ek64FyMWLGh2c5
                        HogVxtt7s9cHICosxqhqZNXYT7GP+YpRRVO4
                        uCGgq6uoqCpgj1L39tqnSQ1da8pT5a6DRCIJ
                        fqsS5ubrmA/20cc02c15XFTlAik= )
            86400 NS extdns1.ipamworldwide.com.
```

```
86400 NS extdns2. ipamworldwide. com.
86400 NS extdns3. ipamworldwide. com.
86400 RRSIG NS 5 2 86400 20100305135354 (
        20100203135354 14522 ipamworldwide. com.
        qVdOx6s9IAL4YWz2hPB1Q5aVNPcPbIsREenD
        PP/7GyXbQKxAdDDugaWPHoKEvPA9f1SBWomZ
        h4pGOKJaA5Pk9okF3FkHLHclTFVGfhTEdrVj
        Dk6a8eRNoU + CMHWwmfJtNFpYpVVd6Ch1LWdw
        ZJ27Z80HZrHtwZ8XmubPzu8MZlE = )
86400 MX 10 smtp1. ipamworldwide. com.
86400 MX 20 smtp2. ipamworldwide. com.
86400 RRSIG MX 5 2 86400 20100305135354 (
        20100203135354 14522 ipamworldwide. com.
        dR4kJtp5DyvCHTF7 + uCNloKCRNVx5jM/XOd9
        H5F7OhnDUIgPWKYnuCbL3PBhx1iK9OnrrL1g
        ZvEuTAvifzzax4n8CSPCB0CbrMWWUXQ44vKG
        I0W0LwzQKJXlPGHzGiG + 6dktfqOnBgppXekA
        QWBJA6nOAeGKtqQMtKUa75uqs2Y = )
10800 NSEC eng. ipamworldwide. com. NS SOA MX RRSIG
        NSEC DNSKEY
10800 RRSIG NSEC 5 2 10800 20100305135354 (
        20100203135354 14522 ipamworldwide. com.
        WyZl4AduBUWdED01Ckc + I0nSArek5n3r6rKX
        m26H5Sjow/RSpgmPJfGOH/9gjyEwnGoqrKbh
        5s7kxtnvF3xVYFE1If7zv5bHxSvBqMDqdNXq
        ChY9BJ9kOemQ0L7NlpreadXfyVXBthl5jaPC
        vKLSwAjmNAzbtV4f6S + CIDK288w = )
86400 DNSKEY 256 3 5 (
        BQEAAAABx6QAwJiz6hHa/eUI2pGz6rvwEYpJ
        di1TJH8Uj4lDPTmzseCOgFEqB3/dZB0Q5LEs
        1ZetAJJEk4F + WccRKwqnIcGkvIKfTC8hn + gb
        iBAnadQRFLxNMBs6KB0e + yqiNK60sbrn22F8
        AYRiG3n2rTQndVtkaZep9jbcCqfu/DagB10 =
        ) ; key id = 14522
86400 DNSKEY 257 3 5 (
        AwEAAdSAwGoUBhtjpE8GLGN4ryt8yEq71Dqd
        E + ij3boe9lmvpM02YZ1/AQxoHbyA7NqRr + 8d
        sTM8OrF2yFRbcP1y0/9q37T0PqxL5HjAZ8Hr
```

```
                DoW9R/pC3XyRe9pMzRNr4as+c/xEISfhxzvR
                84CndF5XvFeh3H0kVDeTb+7QRrG7hnph4P8w
                4SMg76tBvxHLFmj3OdP8vIUpRAnexEAdclam
                j1ZSPjLcdICzpDvQB/LLsYxx8wx2h0vTvhxZ
                klqmy1dPBtIZu2A551VIrU0xgCJxDjJGCgBb
                rp1C01tYSdqlA1I2HCL8eV7io/CxnCuSThPl
                XaPLySojJpXUgDomWgVYeo0=
                ); key id = 6082
        86400 RRSIG DNSKEY 5 2 86400 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                V0bEwZmY56OrGQb02B/Pf17RAcFyPZAvPT/W
                Rm/+nluSOYMVqdzRaKM/ae47KslioXm3tNcy
                GF3uBvBql7xPzIOuIy3COoorXmbsshbuANo7
                YfQsyXWuX2BIjjLAVRRLQolVcdDyyleoA0E7
                BebPM+fQQtvN2C2IjrcacJyeUlc= )
        86400 RRSIG DNSKEY 5 2 86400 20100305135354 (
                20100203135354 6082 ipamworldwide.com.
                e8jCEVY6C1lSImGqjgzVWAgp7cC4AWuntFvc
                oCCO+2BwGxe7+zxP2r02CCSOCIrTqtgwpNRd
                5aH4xBrYmZh0IFQ7OxTFSGBvQ4DxC8ZDdQVS
                uTYCBSzN7kXRJZopZv3chhf7/9uyz3gqtQnl
                5RyUVATMOG5eu+ewBFqGIsXJv5XMNG7ZTO15
                rtRd8zF/7MIY7TlSbHULGP7OJxcNFtyt8wnc
                /dObfcxril4tOwLPVF4QnLnLxAHvdWt+QPVQ
                z23WIc0U+rg6U6FsSjoi0U2QAxVFebenTJED
                U2juAdqEE8I1Y9oOvNQVtYFFjXFgi1vDLGCG
                zM8i4fI9uGZUHvzKng== )
ns1.eng.ipamworldwide.com. 604800 IN AAAA 2001:db8:4af0:2007::7
ns2.eng.ipamworldwide.com. 604800 IN AAAA 2001:db8:4af0:2009::12
eng.ipamworldwide.com. 604800 IN NS ns1.eng.ipamworldwide.com.
        604800 IN NS ns2.eng.ipamworldwide.com.
        10800 NSEC extdns1.ipamworldwide.com. NS RRSIG NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                dWwY0rZRfW5aYgBsbRuCxot6CGGG8hfgHId7
                84IZIYi9HHgr02saBdlzmzqJCGre0pGSDBvf
                ZpJP1BVUS1NuMycEBFBUIS8IUASDTxcLGjrT
                169vIqiyXjICzrsu2fzKL1QNwUOFMGiedglh
```

```
                1jkUJ1jKKs9yr4XFZBwP/y8OpoQ = )
extdns1. ipamworldwide. com. 86400 IN A192. 0. 2. 34
          86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                IwNfRz7m6Rneh6hpacdIpTHGRftsU8e931OP
                bjC0Dfw92DXn51uHghiCoE + rrO4zK1wYFP5L
                CoKF43whVX1EXOt7UFGuAebr4587DnDqhKol
                9XivKc35HvPz1ErniZHuUIsZCjvuziwvGIXS
                72PkoHzNw/lxv + nDriemFn7tWxE = )
          86400 AAAA 2001:db8:4af0:2010::a
          86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                aNzJgdLi4DTttIUj + Y + 9FLI2eAu5iRX9yewN
                jvFG3aJ4moO4fWwhKFynltcfJFpKjHyq4eCD
                PamIS/9fDOn8OdX1g8CkfKNQIszUoAkhSQXH
                6avko1jwgP0lqHwjRNhdcW2UuE + pjyvgNlTW
                ZOgb65nR + UjSJQXRQnHpyhyD + nk = )
          10800 NSEC extdns2. ipamworldwide. com. A AAAA RRSIG
                NSEC
          10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                ipB8eo8GLPvbCCzUF6ETXBiXsRXZiWu8y21z
                uEoxJn + 3T9dYXFEFFpdyj5Qnhl/gnvwpc1mP
                sFyg0 + P5mNziXO/Aj3LQF2HJMnQxT34dQdJb
                Ze/6KBJZO6KZXwXrQXxVGrbFHY9xY5Q0gfs4
                J2MUAZB074KWOVZKUzLUczgrwhI = )
extdns2. ipamworldwide. com. 86400 IN A192. 0. 2. 42
          86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                ax6Umlog3DSn + KxIQSvbQjES9CwuaYZ + G0yT
                NHOIwVOrV4cjP7LA2Pc2p7bQjwoTMkXK5uoU
                Or8Mnd7/boJyQUrBF62pbhOqJ9mKbvrYD1ud
                SivEiDnxAv0FTwagCe22Vvd3DNTjXUhizBt7
                DlIbA92lSCiNHqeFT/OljqcW + Z0 = )
          86400 AAAA 2001:db8:4af0:2011::11
          86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                aQO0ipvwjtAS0DZiXJoTot9iPAToI5rqrkMD
```

```
                lXRNimxuT/ED0+S94OUg5rA5a/XS80aDFSyD
                uqLIVIiZC4Zd5jHazPxEjJR7YyJ0sx8kIy5Q
                85LBJQhVsiADcoKz7NZ8TRFzSEGQNKMLVYIx
                kVx8JpJcGWLeXBekk5J46OeacfE= )
        10800 NSEC extdns3.ipamworldwide.com. A AAAA RRSIG
                NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                J8j82DSNwUc0M2dd2vPzkTlOnjxrrTeKIWH2
                h13hjbH3xr18WLQdJQiqJpXapXSKGX/57+C8
                EO+OBbsqNMpwf+bNhxdnJazB7elYdk7KI8Xp
                TmpyV9zRTJjr3U3l6pw2GjaCMkBDw8JD1+6w
                LJjib4JgHg3pDswvo6ShXxpnezk= )
extdns3.ipamworldwide.com. 86400 IN A192.0.2.50
        86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                a6IVQOXfc0UgsIfCJA/yGDvPdXUrXH2HJzS9
                h/DGEIdu3ZBNcEwtKVvd4ph/rHXknX2Ito2m
                4/1OLtvFdriZjhbpIERCatl45ySxhvugbZlb
                EAjEWa1kixmPoOtXZ+pAS+7cLCxkodr5Np2t
                f9Ppdv5bx4/a9BfM8abrUwrT988= )
        86400 AAAA 2001:db8:4af0:2006::9
        86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                AMeurMSeauKG/w0KSgo9tKWToMDXEOtArCmu
                l3VKDUDN22Y7yfIUX+nwcUJuLRU4tLfeiLBT
                E8IIjsJ3Qu9SQmCBB/4VCHjNax98c4+/RBym
                M9sKuprQK9MEzV5kqqYyHdVuPFzSWCp0QXCO
                AWrWGWfkO3oXS6oj+gqK3hHnAsQ= )
        10800 NSEC ftp-support.ipamworldwide.com. A AAAA
                RRSIG NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide.com.
                nfQMcp6s2IyVItiCmb89DiSKYmdurlBo0Nx3
                0IQYcoMvZVVXMa4ynCoq3lKdebjhGrW8e6NG
                c5SyPYrBzjw1NVEPIr1mNoVN2EEBqquPYluC
                z9f0M5N534yThP01yCsjee7FpIXGKYObhb5+
                i5wLH10NrIpLJEAw3oWsXNPxkhQ= )
```

```
ftp-support. ipamworldwide. com. 86400 INA 192. 0. 2. 44
        86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                umyIYTUI2YaFXcRp9xATrAK7YnOz/PCbzOSF
                xJLL9CLNzmtdPEvFW7iO9oeC8C + R3WfYafhV
                aWiT/BYPbwqvxHaxRWFJ7hIO87n5PHfAHxyE
                dIr1lLZO5f0IKK8oIgawIyHbE/XeqYHVeZpY
                zJSGGMBiyTI/VGKluud17 + /EDh4 = )
        86400 AAAA 2001:db8:4af0:2011::2c
        86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                A43jiBaDMIhL9KtqP2uE2iG + sn5SZBVhqK9Q
                ChRR0512pZJ5WGPip0KjgcJxaVnMbbBuyM7v
                lzW6G1PerBwtbaX/zi2YnW + O0XyBYGlXjXPC
                bHjM3I7Z07WgHD/I4jrHZVQczUDSmZCJQBIK
                zEYlTt + su4K6EIfxw3uBlrheAAc = )
        10800 NSEC smtp1. ipamworldwide. com. A AAAA RRSIG
                NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                v/LRbW7drv03r + F5XasqZ2bjdGXQ7VP6kvOa
                gt3s/gT5W/c8aLfTeA3lmwwEk3DrNEB9U + MV
                XE9YdI1iLySu8J07hF9qJfSiCSIkZgmf5UDZ
                BUUKifIXZVRHUy8uD2pXP3btZOrhR9CXU5oE
                EfrvaGv7 + + yC + IhRJN7pbg + WEU0 = )
smtp1. ipamworldwide. com. 86400 IN A 192. 0. 2. 36
        86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                lSISPwoCpLdSfWFFjhfuASY72DoA06dMPAic
                5vhRJWQfoUbisWrGt29z7r7S7XYIwgRARURO
                JDUSe93z7TzbjxO4UPDbuheFDYI7r + vDXLj2
                cQgKT4gPJ6UCi2kawWaVbAzPz + ZzV2gfxJfc
                fsjARB5rbNDk1BOO6IDI3pfPYh0 = )
        86400 AAAA 2001:db8:4af0:2010::1b
        86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                Jap9zaU4gWcxHzXmtkK8NtCKGUCE/AdPf + /d
                yWJC5PG7ClildQsxCIhbvgLHdQ0YfFMN5nvd
```

```
                abt3fybBoTtbNATZeBqFDalMnF3IBzyhChA +
                0DC1R27LGk7iyOZ5zsq055ZgROpkBbML3o9k
                M7Y + Lx + nM3j44zj6YoUDsAUvP1s = )
        10800 NSEC smtp2. ipamworldwide. com. A AAAA RRSIG
                NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                x1YlFJQBUhSOTB/T7nrntcaB7x96AK + AAJZT
                787XIryUwg5boDkA5MOGNxAoL6nurtbi3 + 6f
                GLDoG4HYLsEmJlamw9 + IANm1u2yLsg5q2viL
                1ymroI0AlpeXNptDevgZ5 + CiRiRKkNw0 + BZ1
                YCrdNJTBUo8pYfZDxdBpihi87EU = )
smtp2. ipamworldwide. com. 86400 IN A 192. 0. 2. 45
        86400 RRSIG A 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                a + qfAnPTIcI7nBNRhg6BDZrFuQvBbiLZUPXA
                kSXeLNkwtK5bodr + j0nZQqUFsCvHw/Gj2FH7
                7L2ROcDto0QHE9WwKy3AjNtvRGg/GK54uO2v
                A4NEx8C0sgIyWPkIC9Nbndp4bE2zV1r3O4Wr
                UkAGYtD/ZMv79vhB8AsLKyfS + yM = )
        86400 AAAA 2001:db8:4af0:2011::2b
        86400 RRSIG AAAA 5 3 86400 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                jQrvFnE/4JqFSfl6b/GR8j2hv/B + 4XmuslCM
                4P2D6YRYGNhZCeOw4DY3U9fGsg + B8gZii7U +
                Rc9Qe8RyzV + wu8gy + 65uvbSl9sb6zfGOrOp2
                P + ZsAy7R0ZtPjzEdMLAIJdea4LdAgUO9IqNo
                Q5ro79H9GAHptAw2epa + 1XAp + wc = )
        10800 NSEC w3. ipamworldwide. com. A AAAA RRSIG NSEC
        10800 RRSIG NSEC 5 3 10800 20100305135354 (
                20100203135354 14522 ipamworldwide. com.
                gKXAbEocdwlnPIo9YtwLwOatBlamwpQTEM + e
                rKjgibrjYP1ymBiRwOs8lnrfXxCbv6v3ix6Q
                IQcQrzKPugVEaIxUl6kqqH0LsXYwgbixppQX
                Lcn1z9Wfmdv23Z6njQrdR + DmF6aQZPlUaiTn
                SPtE26w59U1rtcyhm2p0vqic5l8 = )
www. ipamworldwide. com. 86400 IN CNAME www. ipamworldwide. com.
        86400 RRSIG CNAME 5 3 86400 20100305135354 (
```

```
                    20100203135354 14522 ipamworldwide. com.
                    PTRpKmkZhBw18c1ZscxIlCL7P23fYgZBsEX9
                    DKrawjkyMZ5B + EqQaGNdsfgvmvirrcxCzv/K
                    MaMaeWBGI4Bb9gykm72thXneud5gHi5zLjsK
                    4uK/r3EdWcurQ89R44Q7pTUOy4yzCdpH/KDE
                    Qjd6P/JrWLnO/WW0gxurwHPFiBE = )
          10800 NSEC www. ipamworldwide. com. CNAME RRSIG NSEC
          10800 RRSIG NSEC 5 3 10800 20100305135354 (
                    20100203135354 14522 ipamworldwide. com.
                    W0jcsvdSJfLuJ6l7fAUdfnat9fd45OE6toO7
                    GHwkjRPWm1Q9C83W0Bgc161gZ4r/q1ZKoE6b
                    FvFOIIxz4NhPiJVb2bUbSL6A1K0vwD4KUGL4
                    ExKfuPMl6/gLOE/Tqczcp/2ETXm5yksOkoPJ
                    ynOMoLpBIyNlHeJwtPBaHlbE + B0 = )
www. ipamworldwide. com. 86400 IN A 192. 0. 2. 37
          86400 IN A 192. 0. 2. 53
          86400 RRSIG A 5 3 86400 20100305135354 (
                    20100203135354 14522 ipamworldwide. com.
                    xjwhHWIy25aOvLP2E1y9aaN6GRcGUxoN4o + P
                    eZ0Wc05zjlDu6o1ZOCXivrbIOP4LVS7pqMX3
                    bg4SQDmzmRDQ0H/ + Q8Fzxbf1UFQNcVB2uhtV
                    6R8DfNwRwIugoL + 33qE2MOrrxWz16JutI2qo
                    vkYogNqDj1MNiiKkoGgmJQmiHYc = )
          86400 AAAA 2001:db8:4af0:2010::25
          86400 AAAA 2001:db8:4af0:2006::5
          86400 RRSIG AAAA 5 3 86400 20100305135354 (
                    20100203135354 14522 ipamworldwide. com.
                    lRoCDp + 0y/HM/xyEdciqO5cDWcRzxmQCwPbs
                    GKrCe + OoYHfTFnSBCAEReY4tneb/HMwYbqxV
                    SRp5oW2FPDi5GZunL7tLp7gF0tF7M9XlJVmi
                    9PDg9wiNzDxw/CgbsN/wbtsRpgbPxQwkACiP
                    eRsNDL3Y5EAxLi24yFw + Qay6uEc = )
          10800 NSEC ipamworldwide. com. A AAAA RRSIG NSEC
          10800 RRSIG NSEC 5 3 10800 20100305135354 (
                    20100203135354 14522 ipamworldwide. com.
                    auNzMg6x34 + oradbjFKoQquKmB8sAmKg44FF
                    8FCuh7FI/mrKNHVuv1YmVNXNK/ZHA1JpVYzH
                    fpe4KxPGh8IcDftEfqd52Z0LsetYeRvxNzxQ
```

sAS + OzClCIiTiEpUNte6siExj7YvhBlPN4e4
pnkzTKPULWat489Juzo2U77XysA =)

无需多言，对一个区域签名极大地增加了区域的尺寸。在给定每条 RRSet 的额外 RRSIG 和 NSEC 信息条件下，它也增加了解析报文尺寸，更别提核验回指[⊖]一个信任锚点的信任链所需潜在增加的额外消息流量了。

回去看看我们对数字签名过程的讨论，原始解析数据当然是图 13-1 的要被签名的“数据”。数据实际上由整个 RRSet 组成，它首先被进行散列处理，之后使用密钥对的私有密钥进行签名。得到的签名由每条 RRSIG 记录的签名字段组成。因此，在我们的签名文件开始部分，我们有我们的原始 SOA 记录，后跟其对应的签名（RRSIG）。之后列出我们的三条 NS 记录。由这三条记录组成这个 *RRSet*，依据后面的 RRSIG 记录，进行签名处理。类似的，MX RRSet 被列出并签名。注意，RRSIG 记录指明签名使用的是 ZSK，依据的是密钥标签字段值 14522。DNSKEY RRSet 本身是由 KSK 和 ZSK 签名的，这由带有相应 KSK 和 ZSK 密钥标签的两条 RRSIG 记录得到证明。通常情况下，KSK 仅对 DNSKEY RRSet 签名，ZSK 对所有区域 RRSet 签名。也要注意，ns1 和 ns2. eng. ipamworldwide. com 黏结记录是不被签名的，原因是这些记录是 eng. ipamworldwide. com 区域的权威记录，而不是 ipamworldwide. com 区域的权威记录。

接下来列出的 NSEC 记录，提供了记录的一个规范排序，用来识别并认证一条不存在资源记录的一个否定应答。这条特别的记录指明下一个属主是 eng. ipamworlwide. com。这条 NSEC 记录也被签名。剩余的每条 RRSet 包括一条 NSEC 记录和 RRSet 签名（RRSIG 记录）。

13.3.4　链接信任链

既然区域已被签名，则您应该确定它在信任链中的位置。即确定父区域是否被签名。如果父区域没有被签名，且新的被签名区域是顶层域，即该顶层域被签名（即 zone apex；例如，在本书撰写时，com 是没有被签名的），代表桩解析器查询的各递归解析器，必须采用区域的公开 KSK（作为一个信任密钥）配置。这就通知该解析器，采用这个密钥签名的区域信息是可被信任的。对于那些解析器，它们信任我们的 ipamworldwide. com 区域管理员的数据，则依据如下内容，KSK 06082 公开密钥可被配置在每台递归服务器 named. conf 文件的相应信任密钥（trusted-keys）语句内。

```
trusted-keys {
"ipamworldwide. com. "257 3 5
    "AwEAAdSAwGoUBhtjpE8GLGN4ryt8yEq71DqdE + ij3boe9lmvpM02YZ1/
    AQxoHbyA7NqRr + 8dsTM8OrF2yFRbcP1y0/9q37T0PqxL5HjAZ8HrDoW9
    R/pC3XyRe9pMzRNr4as + c/xEISfhxzvR84CndF5XvFeh3H0kVDeTb + 7Q
    RrG7hnph4P8w4SMg76tBvxHLFmj3OdP8vIUpRAnexEAdclamj1ZSPjLc
    dICzpDvQB/LLsYxx8wx2h0vTvhxZklqmy1dPBtIZu2A551VIrU0xgCJx
```

⊖　回指是从一个下级点指向一个上级点。——译者注

```
DjJGCgBbrp1C01tYSdqlA1I2HCL8eV7io/CxnCuSThPlXaPLySojJpXU
gDomWgVYeo0 =";
};
```

在递归服务器配置内，我们在 ipamworldwide. com 区域处声明了一个信任锚点或 SEP。注意，对于您希望配置的每个信任锚点，都需要带有相应 KSK 公开密钥的一个信任密钥表项。如我们后面将讨论的，您配置的信任锚点越多，则在每个区域密钥切换（rollover）过程中您需要管理的密钥就越多。采用签名根和 TLD 区域，则仅需要维护一个信任锚点。

自动化的信任锚点更新能力，这种做法降低了信任锚点切换的人工管理需求。对于这样的信任锚点，并不使用 trusted-keys 语句，而是使用 managed-keys 语句。在下面的例子中，我们使用来自我们的信任锚点的公开 KSK 作为初始密钥。最初这个密钥作为信任密钥，但随着 ipamworldwide. com 区域的区域管理员依据 BIND 9. 7 和上面提到的定时能力和自动化处理方法，进行发布、激活、退役和删除密钥，这台递归服务器将保持同步，并按照每个信任锚点的方式维护它自己的当前信任锚点密钥库。因此，当撤销这个初始密钥且激活另一个密钥时，采用新近激活的密钥和这个现在被撤销的密钥的 DNSKEY RRSet 签名，就验证了到新近活跃密钥的转换过程。

```
managed-keys {
"ipamworldwide. com. "initial-key 257 3 5
    "AwEAAdSAwGoUBhtjpE8GLGN4ryt8yEq71DqdE + ij3boe9lmvpM02YZ1/
    AQxoHbyA7NqRr + 8dsTM8OrF2yFRbcP1y0/9q37T0PqxL5HjAZ8HrDoW9
    R/pC3XyRe9pMzRNr4as + c/xEISfhxzvR84CndF5XvFeh3H0kVDeTb + 7Q
    RrG7hnph4P8w4SMg76tBvxHLFmj3OdP8vIUpRAnexEAdclamj1ZSPjLc
    dICzpDvQB/LLsYxx8wx2h0vTvhxZklqmy1dPBtIZu2A551VIrU0xgCJx
    DjJGCgBbrp1C01tYSdqlA1I2HCL8eV7io/CxnCuSThPlXaPLySojJpXU
    gDomWgVYeo0 =";
};
```

现在如果刚被签名的区域，是一个被签名父区域的一个子区域，则父区域管理员必须在父区域文件中包括委托签名者记录，以此链接信任链。采用这种方式，父区域可担保这个被签名的子区域。因此，针对这个区域的信任锚点不需要配置在解析器或递归服务器中，仅配置它的父区域或甚至较高层次祖先的被签名区域。

Dnssec-signzone 设施工具的-g 选项，自动地在一个 dsset-ipamworldwide. com 文件中为区域产生我们的 DS 记录。该文件包含两条 DS 记录，其中一条记录可依据首选摘要类型进行选择。在下面的例子中，在摘要之前所示的整数指明摘要类型。类型 1 是 SHA-1、类型 2 是 SHA-256。接下来是摘要，它是作为一个散列计算得到的，即采用被签名区域的相应摘要类型或算法对 KSK DNSKEY 资源记录属主和 RData 字段（即 KSK DNSKEY 记录，忽略 TTL、类和类型）计算得到的。

```
ipamworldwide. com.
INDS6082515F696637B085D8F5CBFD0C8B9E031CB6CB07159B
```

ipamworldwide. com. IN DS 6082 5 2

7FFD9203E916B5D49F631D060FAFD05D26974BEFCED25AACB88122722E4A7AA9

以认证记录或其签名子区域（委派的）中的不存在性（nonexistence），委派签名者资源记录类型提供了从一个父区域到一个委派子区域的密钥的链接，作为信任链内的一个链接。接下来我们将在解析过程内详细讨论这是如何工作的。DS 资源记录具有如下格式

属主	TTL	类	类型	RData
委派域	TTL	IN	DS	密钥标签　算法　类型　摘要
ipamworldwide. com.	86400	IN	DS	6082 5 1 5F695D8F5BFD0C...

DS 记录的 RData 部分识别子区域的公开密钥的密钥标签或 ID，而算法则匹配引用（referenced）DNSKEY 记录的算法字段。摘要类型指明在摘要字段携带的散列或摘要的类型。有效摘要类型值是 1（SHA-1）或 2（SHA-256）。摘要字段包含对应子区域公开密钥 KSK DNSKEY 资源记录属主字段与相同 DNSKEY 记录 RData 字段串接后的摘要或散列结果。父区域管理员将 DS RRSet 添加到父区域，并对其签名，来认证它的源发性和完整性。

在 BIND 9.6 中，引入了一个新的设施工具 dnssec-dsfromkey。在不需要使用 dnssec-signzone 对区域重新签名的条件下，这个设施工具支持 DS 资源记录生成。这个设施工具带有如下参数。

（1）-1：使用 SHA-1 作为摘要算法。

（2）-2：使用 SHA-256 作为摘要算法。

（3）-a 算法：其中算法如下。

1）SHA-1。

2）SHA-256。

（4）-A：为生成 DS 记录，包括 ZSK 和 KSK；如果忽略的话，则仅产生 KSK 的 DS 记录。

（5）-c 类：识别类（默认的是 IN）。

（6）-d 目录：密钥集合文件的目录位置。

（7）-f 文件：指定一个区域文件名，而不是指定密钥文件名。

（8）-l 域（domain）：生成一个 DLV 集合（而不是一个 DS 集合），并将域（domain）附加在集合中的每条记录。

（9）-K 目录：定义密钥文件所处的目录。

（10）-s：命令参数是一个域名，而不是一个密钥文件名。

（11）-v 等级：指定调试等级。

dnssec-dsfromkey 设施工具可依据一个密钥文件或一个域名，生成 DS 或 DLV 记录；-s 参数将参数定义为一个域名

dnssec-dsfromkey -s [-v level] [-1] [-2] [-a algorithm] [-l domain] keyfile

忽略-s 的做法，则将参数识别为一个密钥文件名。

dnssec-dsfromkey [-v level] [-1] [-2] [-a algorithm] [-l domain] [-K directory] [-c class] [-f file] [-A] domainname

13.4　DNSSEC 解析过程

现在让我们回顾一下解析过程和验证过程是如何工作的。将上面的信任密钥或管理密钥语句配置到我们的递归服务器配置（named. conf）之中，则我们声明 ipamworldwide. com 为一个信任区域。当我们发出对 ipamworldwide. com 区域内一台主机（例如 ftp-support. ipamworldwide. com）的一条查询时，我们的解析器就设置 EDNS0 扩展 Rcode 字段中的 DNSSEC OK（DO）比特。DNSSEC 要求 EDNS0，为的是支持这个扩展的 Rcode 字段，并且一般而言，对于大量的响应报文，看来也超过了标称的 512 字节 UDP 报文限制。报文长度增加的原因是由于服务器的响应造成的，它配置有权威的签名区域，不仅带有被请求的解析数据（ftp-support. ipamworldwide. com 的 A 记录（可能有多条记录）），而且带有与 A 记录集合关联的关联记录。

13.4.1　验证签名

签名过程对资源记录集合签名，这些集合是带有共同属主名、类和类型的资源记录分组。签名是使用由密钥标签参数索引的私有密钥产生的，并被放置在 RRSIG 资源记录的签名字段内。

RRSIG 资源记录具有如下格式。

属主	TTL	类	类型	RData
RRSet 属主	TTL	IN	RRSIG	覆盖的类型　算法　标签　原始 TTL　超期 开始时间　密钥标签　签名者　签名
ftp-support. ipamworldwide. com.	86400	IN	RRSIG	A 5 3 86400 20100305215354 20100203215354 14522 ipamworldwide. com. umyI…

RRSIG 记录内的 RData 字段定义如下。

（1）覆盖的类型。由这个签名所覆盖的资源记录集类型。在我们的例子中，由这个签名覆盖的是 A 记录类型，它对带有属主 ftp-support. ipamworldwide. com. 的我们的两条资源记录 RRSet 进行了签名。

（2）算法。生成该密钥过程中使用的算法，它以 DNSKEY 资源记录类型的算法字段（见前面的 DNSKEY）一样的方式进行编码。

（3）标签数量。指明属主字段内的标签数。例如，ftp-support. ipamworldwide. com 有三个标签。这个字段被用来重构原始的属主名，该名被用来在如下情形下产生签名，其中由服务器返回的属主名有一个通配符（*）。

（4）原始 TTL。依据在权威区域定义的被签名 RRSet 的 TTL，来核验一个签名。需要这个字段，原因是在原始响应中返回的 TTL 字段，正常情况下，会被一个缓存解析器减少，使用 TTL 字段可能导致错误的计算。

（5）签名超期。这个签名超期的日期和时间，表示为自 1970 年 1 月 1 日 00：00：：00 UTC 以来的秒数，或以 YYYYMMDDHHmmSS 形式表示，其中

1）YYYY 是年（在当前日期的 68 年内，目的是防止这个字段的数值回绕）。

2）MM 是月份，01～12。

3）DD 是该月中的日，01～31。

4）HH 是 24h 表示法中的 h，00～23。

5）mm 是 min，00～59。

6）SS 是 s，00～59。

7）在这个日期、时间之后，签名是无效的。

（6）签名开始时间。这个签名开始时的日期和时间，其格式与签名超期字段的方式相同。在这个日期/时间之前的签名是无效的。

（7）密钥标签。以密钥 ID 或标签提供与相应密钥（DNSKEY 资源记录（可能有多条））的一种关联。

（8）签名者的名字。识别 DNSKEY 资源记录的属主名，用其来产生这个签名。

（9）签名。密码学签名，涵盖 RRSIG RData（除了这个签名字段自身）和资源记录串接部分，资源记录是由 RRSIG 属主识别的 RRSet、类和涵盖的类型字段组成的。

因此，对我们的查询的响应，包括 A 记录和关联的 RRSIG 记录，由这条响应指明，响应是由如下的 dig 设施工具捕获到的。服务器将在响应中的 DNS 首部中设置认证数据（AD）比特，仅当它已经认证了（以密码学方式验证的）所有包括在答案节中的资源记录以及所有包括在权威节中的负面响应资源记录。注意，如果您查询了作为所发出查询的权威服务器，则 AD 比特将不被设置。这台服务器简单地返回答案，并将验证工作直接留给查询器处理。如果您查询您的递归服务器，它不是所查询信息的权威服务器，则它将实施解析和 DNSSEC 验证，如果验证成功，则将在结果中设置 AD 比特。我们将在第 14 章回顾 dig 设施工具的细节，在验证并排查区域配置问题中，该工具是非常有用的。

```
$ dig  + dnssec A ftp-support. ipamworldwide. com. @ 127. 0. 0. 1
; < < > > DiG 9. 6. 2 < < > > + dnssec A ftp-sf. ipamworldwide. com. @ 127. 0. 0. 1
; (1 server found)
; ; global options: printcmd
; ; Got answer:
; ; - > > HEADER < < opcode: QUERY, status: NOERROR, id: 462
292 SECURING DNS (PART I I) : DNSSEC
; ; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 3
; ; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
; ; QUESTION SECTION:
;ftp-sf. ipamworldwide. com. IN A
; ; ANSWER SECTION:
```

ftp-sf. ipamworldwide. com. 86400 IN A 10. 1. 32. 9

ftp-sf. ipamworldwide. com. 86400 IN A 10. 1. 32. 5

ftp-sf. ipamworldwide. com. 86400 IN RRSIG A 5 3 86400 20100525173519

20100425173519 14522 ipamworldwide. com. owHoS6b1xTNKuzJjgJs3nL4Kwr-

LehnfixVjAF2T6 RHu4dVmq4w1p + FNC Oji2BkWKOhjY3 + 7jU4doFr/RNioe8vmsqyn

R5YeSSRzzFy/d63Riz3bQ5BANbGRqpTn6Q9HQlm + KYSpwY5CrjqOQnP + Ynme4nhT9

+ z8h5ahdwtK9 EtI =

;; AUTHORITY SECTION:

ipamworldwide. com. 86400 IN NS ns. ipamworldwide. com.

ipamworldwide. com. 86400 IN RRSIG NS 5 2 86400 20100525173519

20100425173519 14522 ipamworldwide. com. OLonIvBmJZDEZoRRvOiq7GnlWnr-

8LTWHtKSR60CJNl3hd23Vvkbq/EkV

46wp6OK6Q0qNtJGE + YqFW9xml7d6kQRZOqIyCiDZqHQinV7LlAa0Da8z5 +

UGduD3gVLceES7lvGZpLlbyUm9kFGf5FhPZ/

JciPF4qKUdAvfEeitu/aY =

;; ADDITIONAL SECTION:

ns. ipamworldwide. com. 86400 IN A 10. 1. 32. 4

ns. ipamworldwide. com. 86400 IN RRSIG A 5 3 86400 20100525173519

20100425173519 14522 ipamworldwide. com. IHtLJaWam57mVoYCgFqlEPC9N9p7n-

Wicy7MBvdQP6PgNfhnOTOg2vQHR rQRDdBWBmgaSRoiWSdF2IQTEfH4T16591-

OEjtBnPR/7zRAxU9abnkUDvGCZsAFfqKfWxBZFrxUTbxloekEhMC98FqCnvaRIsLNY-biP/

0KhehWmBF nlA =

已经接收到 RRSet（数据）和 RRSIG（签名）的递归服务器或解析器，如果 DNSKEY RRSet 没有被缓存或在响应的附加节中提供的话，之后递归服务器或解析器会发出一条 DNSKEY 查询，以便得到 DNSKEY RRSet。采用标签 14522 的密钥处理 RRSIG 记录中的签名，并与 RRSet 内和资源记录串接在一起的 RRSIG RData（去掉签名）的散列比较。如果比较得到一次匹配，则签名被成功核验通过。接下来，DNSKEY RRSet 的 RRSIG 被用来核验 ZSK 本身。和刚刚描述核验 A RRSet 一样，解析器或递归服务器实施一种类似的计算，就公开 KSK 签名来核验 DNSKEY RRSet。考虑成功的匹配和这样的事实，即公开 KSK 匹配一个配置的信任密钥，因此我们就成功地核验了 RRSet 数据。

13.4.2　经过认证的存在性拒绝

如果我想查询一个主机名，但我敲错了，会发生什么呢？在没有 DNSSEC 条件下，我将接收到一条错误（NXDOMAIN）信息，指明记录不存在。为了解决这种潜在的弱点，DNSSEC 集成下一条安全（Next SECure，NSEC）信息资源记录，提供匹

配查询的一条记录不存在性认证的一种方法。本质上而言，NSEC 记录在区域文件内从一个 RRSet 指向下一个 RRSet，识别 RRSet 之间的间距。NSEC 记录的格式如下。

属主	TTL	类	类型	RData
RRSet 属主	TTL	IN	NSEC	下一条 RRSet 属主 类型比特映射
ns1. ipamww. com.	86400	IN	NSEC	ns2. ipamww. com. A AAAA RRSIG NSEC

在这个例子中，与属主 ns1. ipamww. com 关联的 NSEC 记录，指明以规范顺序表示的下一个属主名是 ns2. ipamww. com，这个属主名（ns1）与类型为 A、AAAA、RRSIG 和 NSEC 的资源记录一起存在。这条记录指明，在 ns1. ipamworldwide. com 和 ns2. ipamworldwide. com 之间不存在规范化的任何记录，例如 ns1a. ipamworldwide. com。每个 NSEC RRSet 也采用私有 ZSK 签名，私有 ZSK 接下来和一个信任 KSK 进行比对核验。

NSEC3 提供 RRSet 的类似经过认证的存在性拒绝指示，但它也在区域中混杂 RRSet 的平凡枚举，这被认为是一项信息安全风险。因为 NSEC3 记录使用一个精选值（salt）和经过散列处理过的属主姓名，这进一步使散列字段产生函数复杂化了，要枚举该区域，从计算角度看代价是较高昂的。如此，当对区域签名以及当对指明记录不存在的查询解析进行核验时，计算代价也是高昂的。

13.4.3 在一个信任链中的父区域委派

现在让我们扩展我们的例子，来形象地说明生成到一个信任锚点的内部区域信任链中 DS 记录的角色。考虑图 13-3，其中我们从解析的数据到信任锚点一路仔细研究。让我们假定 ipamworldwide. com. 区域（密钥 ID = 06082）的公开 KSK 被配置为我的递归服务器中的一个信任锚点。当我发出对 host. child. ipamworldwide. com. 的一条 A 记录查询时，名字解析遵循传统的域树遍历过程，以便得到一个缓存的或权威的答案。假定我设置了 DO 比特，则返回解析 RRSet 和对应的 RRSIG 记录。递归服务器可采用 child. ipamworldwide. com 的 ZSK（密钥 ID = 98211）核验 RRSIG，如图 13-3 中标记“1”的箭头所示。接下来，依据步骤 2，可采用区域的 KSK（密钥 ID = 45443）核验 ZSK。考虑到我没有将这个 KSK 配置为一个信任锚点，所以我不能信任这个数据。但是，递归服务器查询父区域 ipamworldwide. com，查询一条 DS 记录，来确定是否父区域可认证这个区域的数据。这个过程如图 13-3 中的步骤 3 所示。

如果 DS 记录摘要匹配相应的 child. ipamworldwide. com 区域的 KSK DNSKEY 数据，则我就能够得出结论，即 ipamworldwide. com. 区域已经对 child. ipamworldwide. com 的委派进行了签名。之后递归服务器将 DS 记录上的签名与 ipamworldwide. com 的 ZSK（密钥 ID = 14522）核验，接下来是它的 KSK（密钥 ID = 06082），后者被配置为一个信任密钥。因此，通过反向到一个配置的信任锚点的信任链，我们确认原始数据被解析为信任的。对于沿域树向上到签名根域的任何数量的父区域-子区域重复，可使用根区域信任锚点，重复这个相同的过程。最终我必须在我的查询可施用的区域或其祖先区域之一配置一个信任密钥。考虑各种区域（包括反向区域），它们需要对查询进

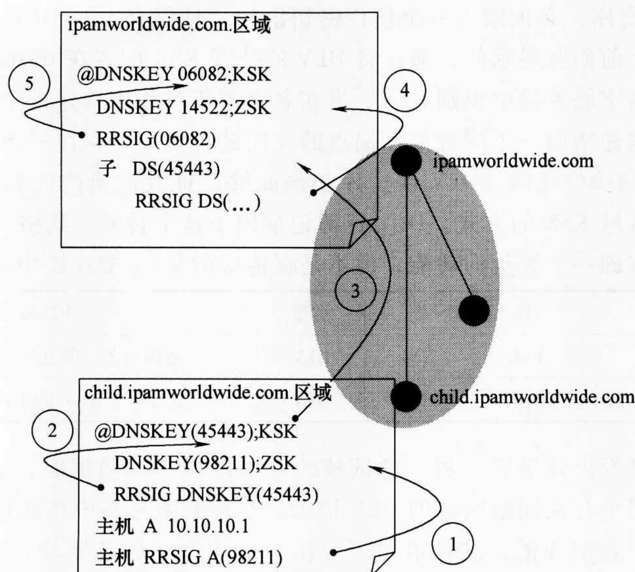


图 13-3　DNSSEC 信任链遍历

行认证，这个信任锚点集可快速地变得非常巨大。

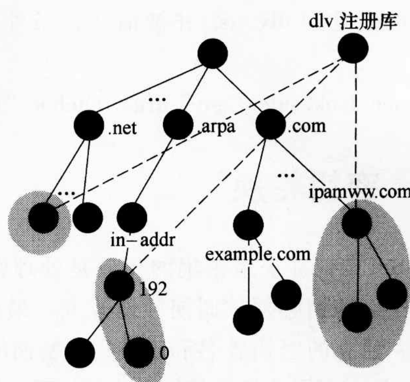


图 13-4　DLV 信任链

DNSSEC 旁查核验，作用为在根区域被签名时间之前，有助于保持信任锚点集合在一个可管理的水平。DLV 利用签名区域公开密钥的一个中心式注册库（registry）。通过将 DLV 注册库配置为一个信任锚点，由此您可信任 DLV 注册库和所有它认证的“子”区域。这些区域不是 DLV 的真正子区域，而是 DLV 认证的区域。区域管理员以一种安全的方式将它们签名的区域密钥注册到 DLV 注册库，以此维护这个“旁查”或“旁路”（sideways）信任链，这与我们刚讨论的域树父-子信任链截然不同。

图 13-4 形象地说明了这个概念。在没有根和 TLD 区域签名的条件下，不得不针对每个信任区域配置信任密钥。在图中，这些被表示为 ipamww. com、192. in-addr. arpa 和一个 . net 区域。通过使用 DLV 概念，DLV 对 DS 等价的 DLV 记录进行签名，目的是认证每个“子”区域的 KSK。一个 DLV 注册库的优势是在您的组织机构中的每台递归服务器中降低被管理的密钥数量。如图 13-4 所示，如果 DLV 对三个区域密钥签名，则您仅需要关注 DLV 的密钥轮换（rollover），而不是作为组成部分的这三个密钥。必须完全地信任 DLV 注册库，原因是它所认证的区域不是为注册库用户可选择接受的（即必须接受）。

针对 DLV 注册库，必须输入一个信任密钥语句，且仅有一个 DLV 注册库可被如此引用。如我们在前面所展示的，通过将 DLV 的公开 KSK 配置在 trusted-keys 语句块的做法，在递归名字服务器中识别 DLV。当在名字解析过程中构建一个信任链时，递归服务器将尝试构建指向一个配置信任锚点的反向链；如果不存在一条有效的链，它将尝试通过 DLV 核验信任链。DLV 注册库必须能够认证其注册的区域，这很像一个父区域核验其子区域 KSK 的方式。DLV 资源记录用于这个目的，其格式等同于 DS 资源记录类型。它实施一个等价的功能，但不是在传统的父-子委派链中执行。

属主	TTL	类	类型	RData
DLV 域	TTL	IN	DLV	密钥标签 算法 类型 摘要
ipawmww. com. dlv_reg. net.	86400	IN	DLV	32284 5 1 90df80DF89lLe. . .

当递归服务器发出其最后一招，尝试核验一个区域中的数据时，它在 DLV 注册库中寻找对应于那个存在问题区域的 DLV 记录。在递归服务器中是通过 dnssec-lookaside 语句识别 DLV 注册库的，这种语句配置在 named. conf 的选项块内。这条语句识别提升到 DLV 注册库的域树分支是有效的，以及到信任锚点的一个索引，该信任锚点是在 trusted-keys 语句中识别的。例如，下面的语句表明，在 gov. domain 内的解析可被提升到 dlv. us. 并被信任，条件是 dlv. us 公开 KSK 匹配配置的 dlv. us 信任密钥。

dnssec-lookaside “gov” trust-anchor “dlv. us”;

13.5 密钥轮换

DNSSEC 管理上最密集的任务是处理密钥轮换过程，特别是 KSK 轮换过程。就像密码一样，密钥必须被周期性地改变。最好的方法，是向可能的攻击者提供一个移动的目标。独立的密钥签名和区域签名密钥的用法，有助于这个过程的管理。这是由于如下事实，即在不影响任何其他人的条件下，任何一名区域管理员均可使用一个 ZSK 简单地对他的/她的区域重新签名。无论何时使用 ZSK 时，它最终都是采用 KSK 签名的，KSK 可被配置为一个信任锚点或由一个 DS 或 DLV 资源记录索引。因此，ZSK 可依据意愿进行改变。但是，因为各 KSK 被配置为信任锚点，并可能为其他区域的 DS 或 DLV 记录索引，所以它们确实会影响其他管理员，并要求采用一个非常严格的集成过程。

用于密钥轮换的两种基本方法是预产生（preseeding）密钥（对应 ZSK 轮换是有效的）和双密钥签名方法（可被用于轮换 KSK）。密钥发行（不是有意的双关语（指 issue））和轮换与递归服务器和解析器中被缓存解析和签名信息的更新有关。当一个解析器得到认证的解析信息时，它将在原始记录 TTL 的时间段内缓存这个信息，包括含 ZSK 和 KSK 的 DNSKEY 记录。在 TTL 超期之后，解析器必须发出请求相应信息的一条新查询。如果一名区域管理员实施一个新密钥替换一个旧密钥的瞬间切换（flash cut），则以旧密钥（依据其 TTL 仍然有效）实施查询的解析器和递归服务器，将不能够认证该区域内使用新密钥签名的所解析数据。图 13-5 形象地说明了这种时

间影响。因此，维护一个窗口，在其间密钥更新可被传播且两个或多个密钥都是有效的，这构成基本的轮换技术。

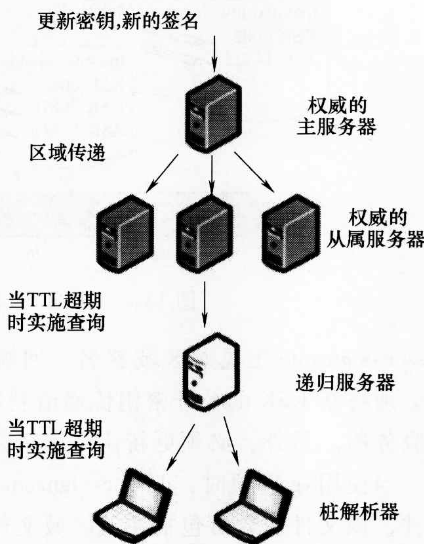


图 13-5　区域信息传播

让我们首先考虑图 13-6，讨论和比较两种常见的轮换策略。如果，如果可能的话，ZSK 和 KSK 轮换应该是独立发生的。我们将假定预产生策略被应用到 ZSK 轮换，而双密钥签名方法被应用到 KSK 轮换。首先研究 ZSK 轮换，我们的初始条件是，一个区域以 ZSK［密钥标签］14522 和 KSK 6082 签名，由笔形图符指示。在时间 t_0，预产生时间为 1s，产生“被动的”ZSK 28004，使用的是 dnssec-keygen 工具或通过 BIND 9.7 + 自动化方法，其相应的 DNSKEY 资源记录被包括在区域文件中，还有主动的 ZSK 14522。在区域文件中插入或包括 ZSK 28004 之后，必须对区域重新签名，仍然使用的是 ZSK 14522。被动的 ZSK 本身使用主动密钥签名，并可由解析器和递归服务器缓存，但还没有用来对区域数据签名。

一旦发布的话，这两个密钥都应该保留在区域文件中，直到所有从属服务器通过区域传递得到区域文件，以及密钥超期时间之后，才可被去除。密钥超期时间应该比区域或资源记录 TTL 要长。当在时间 t_1（轮换时间）过了这个时间时，对该区域重新签名，这次使用的是前面的被动 ZSK 28004。在等价间隔期间，直到时间 t_2 之前，前面的主动 ZSK 可留在区域之中，之后会从区域文件中被去除。取决于 ZSK 轮换的频率，时间 t_2 可对应于下一个密钥轮换周期的 t_0，其中区域将总是有两个 ZSK，一个是主动的、一个是被动的。否则，在这个时间点，仅有主动 ZSK 存在于区域内。

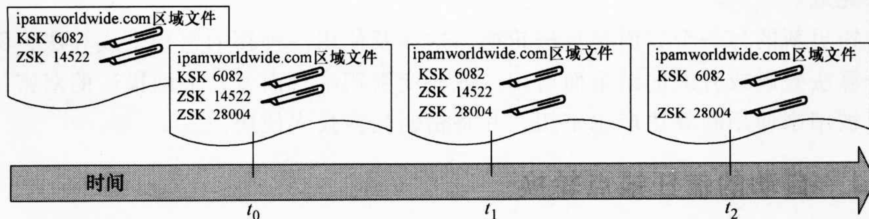


图 13-6　DNSSEC 预产生密钥轮换策略[11]

现在让我们详细研究双密钥签名轮换方法，如图 13-7 所示。这个过程开始时与我们前一个例子中的初始条件相同。在轮换时间 t_0，使用带有-k 选项的 dnssec-keygen 工具生成一个新的 KSK70810。现在利用当前 KSK、新的 KSK 以及主动 ZSK，使用

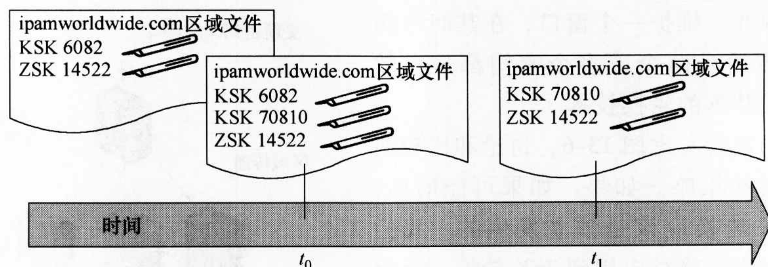


图 13-7 DNSSEC 双密钥签名密钥轮换策略[11]

dnssec-signzone 工具对区域签名。回顾一下，dnssec-signzone 允许指定多个 KSK。之后必须将新 KSK 的公开密钥传递给利用这个区域作为一个信任锚点的所有解析器/递归服务器。另外，必须更新认证这个区域的父区域。

当使用-g 选项时，dnssec-signzone 工具的一个输出包括一个 dsset- <zonename> 文件，该文件包含可包括在父区域文件中的相应 DS 资源记录（可能是多条记录）。-l 选项生成一个 dlvset- <zonename> 文件，该文件包含相应的 DLV 资源记录。父区域或 DLV 管理员必须复制或包括这些 DS 或 DLV 记录（分别情况处理），并对父区域重新签名。考虑到在父区域和解析器/递归服务器上实施这些任务所要求的人工配置，相比于预产生方法，这个时间帧是不太确定的。一旦这个时间消逝，且父区域和信任锚点配置被更新，则旧 KSK 可从区域文件中去除，并仅可使用新近的 KSK 对区域重新签名，如在时间 t_1 所示情况。

在出现对应于一个主动 KSK 或 ZSK 的一个私有密钥被破解情况下，应该设计紧急轮换过程。如果一名攻击者得到私有密钥，他/她会伪造区域数据，并使用私有密钥对其签名。解析器和递归服务器将依据对应的发布的公开密钥，认证伪造的数据。如我们看到的，ZSK 可被强制性地改变，所以此时应该立即改变 ZSK。但是，改变 KSK 确实要求比较广泛的参与和协作。我们建议对紧急轮换的过程形成文档，它包括父区域管理员和 DLV 注册库联系方式，以及联系用户的一种方式，这些用户将该 KSK 配置为一个信任锚点。这可通过一个注册（registered）电子邮件列表和安全网站发布来完成。

密钥更新的另一个方面是算法轮换。这涉及使用一种新的密钥生成算法，例如作为一个算法破解或升级的结果而启用。就改变密钥本身而言，上面描述的双密钥签名过程可被用来使用新算法产生密钥，并将密钥轮换投入使用。

13.5.1 自动的信任锚点轮换

RFC 5011[143] 定义了自动化信任锚点轮换的一种方法，目的是降低在所有解析器/递归服务器上更新信任密钥的管理方面的影响，这些解析器/递归服务器使用这个区域作为一个信任锚点。这个自动化方法要求针对信任锚点区域，进行当前公开 KSK 的初始配置。但不像人工配置的是，在每次发生信任锚点 KSK 变化时，不需要人工的更新。可在 BIND 9.7 及以上版本中使用 managed-keys 语句，在一台递归 BIND 服

务器中配置自动化的信任锚点更新。初始信任密钥将被用来核验使用 DNS 协议传递的未来密钥事务。解析器必须周期性地查询信任锚点，查询其 DNSKEY RRSet，以便检查更新。如果一个新的密钥添加正确，它将自动地被认为是该区域的一个有效信任锚点密钥，条件是被当前信任密钥签名。如果当前信任密钥被撤销，且由区域的信任密钥（可能是多个密钥）签名，则信任密钥将自动地从处理中被清除。因此在 managed-keys 语句中配置的初始密钥，仅被用作信任锚点初始条件；这个密钥可在未来被撤销，且 DNS 服务器自动地跟踪当前信任密钥的状态。

图 13-8 给出解析器角度来看的信任密钥的一个状态图，依据的是通过核验过的（由当前信任密钥（可能有多个密钥）签名）DNSKEY 查询检索到的密钥状态。当在 DNSKEY RRSet 内的服务器检索一个新的 SEP 密钥（信任锚点）时，密钥进入 Add Pending（添加进行中）状态。这个状态有助于缓解如下情形，其中一名攻击者已经破解信任密钥，并寻求使解析器信任来使用攻击者的新密钥。如果在 add hold down（添加抑制）定时器过程中的任何时间，解析器在 DNSKEY RRSet 中都没有看到进行中的密钥，则将认为该密钥是无效的。在这个间隔过程中，一名攻击者要对每次 DNSKEY 查询都正确地做出响应，将是非常具有挑战性的。一旦抑制定时器过期，则信任密钥进入有效状态，并被认为是区域的一个有效信任密钥。在这个状态，如果密钥从 DNSKEY RRSet 中丢失，则将被认为是丢失了（Missing），但在重新出现时，会以有效的签名将其重新恢复到有效状态。

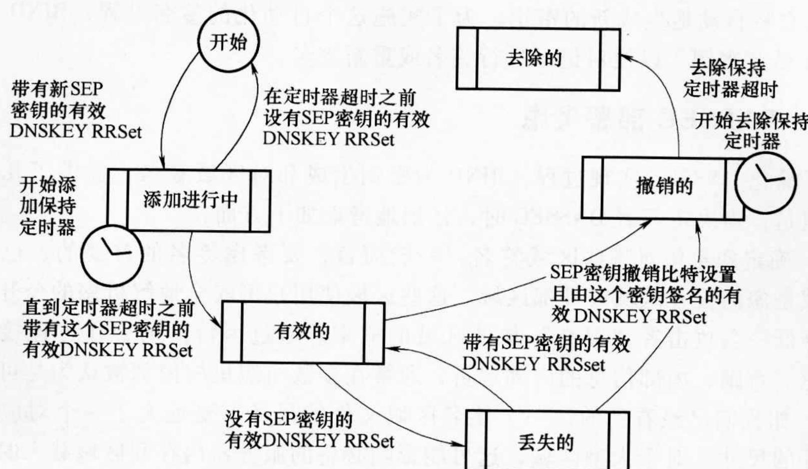


图 13-8　信任锚点（SEP）状态图[143]

当由于密钥在其寿命中的年龄（到期）或因为密钥被破解，区域管理员期望撤销该密钥时，该密钥将以 DNSKEY 标志字段中设置撤销比特的方式发布在区域之中。除了任何其他主动的或进行中（pending）的信任密钥外，这个密钥必须被用来签名 DNSKEY RRSet。在这种情形中，该密钥将被认为被撤销了。可从有效状态或丢失状态，进入这个状态。服务器启动一个消除抑制定时器，当超期时，激发从服务器配置中消除信任密钥的操作。

13.5.2 DNSSEC 和动态更新

区域签名过程要求一个区域的规范排序，之后才是签名，在这种情况下，您可能会疑惑，即人们如何将一条新的资源记录安全地插入到该区域中呢？幸运的是，区域签名并不要求对整个区域进行重新处理，而是对个体 RRSet 签名的，这种做法使一个比较模块化的过程成为可能。但是，为考虑更新，必须调整 NSEC [3] 记录，方法是调整规范排序，才能有效地插入更新。

当动态地更新一个安全的区域时，更新本身必须是安全的。服务器应该要求对更新消息进行签名，并应该定义哪些服务器或网络会实施更新。当接收到一条更新，并认证通过时，它被保留在日志文件内。为了对带有更新的区域完整地进行签名，服务器必须临时地冻结动态更新，当使用 pre-BIND 9.6 时，使用的是 rndc freeze 命令。这就关闭了动态更新的接收。一旦冻结，则必须使用 dnssec-signzone 功能对区域重新签名。之后，可使用 rndc thaw 命令重新激活动态更新。

在 BIND 9.6 及以上版本中已经消除了这个人工冻结签名解冻（freeze-sign-thaw）过程，在这些版本中为动态更新添加了一个自动地签名机制，这极大地简化了这个过程。与其将正常的日志更新集成到区域文件一起，BIND 使用 ZSK 以及相应的“之前”和“之后”的 NSEC [3] 记录，对每条更新签名，从而规范地将记录插入到区域之中。在签名接近超期时，BIND 9.6 也周期性地检查区域的签名。在这样的情形中，那么它将自动地生成新的密钥。为了实施这个自动化的签名过程，BIND 必须可访问 ZSK 私有密钥，以便对记录进行签名或重新签名。

13.5.3 DNSSEC 部署考虑

为了简化 DNSSEC 实现过程，BIND 为密钥生成和对区域签名，提供了几项工具设施。但是，当决定部署 DNSSEC 时，仔细地考虑如下方面：

（1）确定您希望对哪些区域签名。一般而言，要考虑签名的首要的、也许是唯一的区域是您的公开区域或外部区域。这些区域使用户可安全地解析您的公开名字空间，并降低一名攻击者“冒充”您的区域的概率。通过因特网的合作方连接应该做类似考虑。否则，内部信息的内部解析，通常在多数组织机构内部被认为是可信的。

（2）如我们已经看到的，一个签名区域文件的尺寸，要远大于一个对应未签名区域文件的尺寸。对于大型区域，这可能影响必备的服务器内存和区域载入时间。

（3）对您的区域签名的做法，保护的是您的名字空间的完整性，而不是您的 DNS 缓存的完整性。考虑在您的因特网查询 DNS 服务器上配置 DNSSEC 核验。

（4）考虑到附接上 RRSIG 记录以及对应于查询的可能 DNSKEY 记录，一条给定查询的解析响应也会增长得较巨大。这可能负面地影响查询响应时间和性能。

（5）解析过程性能也可进一步受到信任锚点确认过程的负面影响，其中密钥和委托签名者记录要被核验，可能会到达信任锚点区域或 DLV 注册库。

（6）DNSSEC 引入对时间同步的要求，其中给定绝对时间参考，在 RRSIG 记录中指明有效时间和超期时间。

（7）通过跳过 NSEC 记录的区域踪迹法，是一个潜在的信息过度暴露，虽然 NSEC3 记录使这个过程比较困难。考虑区域踪迹法对您而言是否真是一个问题（一般情况下，在 DNS 中发布的信息是公开信息），这源于在一个签名区域内产生 NSEC3 记录的计算复杂度和潜在的大量时间。

（8）用于初始化和轮换的密钥更新过程，必须设计成如下方式，即如果您的信任解析器没有使用自动化的信任锚点更新特征时，通过一种带外机制提供对更新 KSK 的认证访问。KSK 公开密钥更新，必须被传递到您信任的所有区域以及您的父区域或 DLV（如果存在的话，则分别采用 DS 或 DLV 记录的形式）。

（9）DNSSEC 实施数据源发认证、数据完整性验证和经过认证的存在性拒绝。它不能对第 12 章介绍的其他弱点类型进行保护。不要忘记，在那一章讨论的实施缓解战术以便防护其他弱点。

第Ⅳ部分 IP 地址管理（IPAM）集成

在本书前三部分讨论 IPAM 的组成部件之后，第Ⅳ部分处理这些部件的集成管理任务。我们在第 14 章开始讲述总的 IPAM 技术。之后在第 15 章我们将讨论 IPv6 在一个 IPv4 网络中的实现和共存策略。

第 14 章 IPAM 实践

在本书前言中，我们讲到，IP 地址管理实践（IPAM）包括将网络管理学科（discipline）应用到 IP 地址空间和相关联的网络服务。因为 IP 地址以及相关联的 DHCP 和 DNS 功能，对于运行在一个网络上的 IP 服务和应用而言起到如此的基础作用，所以这些功能必须得到慎重的管理，这非常像其他关键性的网络基础设施单元要得到管理一样。将 DNS 和 DHCP 服务器看作网元的做法，有点超前，这是因为它们对一个 IP 网络上的客户端而言，提供关键的 IP 服务。虽然不像传统网元那样处在用户 IP 流量的带内或数据路径上，但它们提供使这种带内数据路径成为可能和有用所要求的必要服务。从一个电话智能网类比看，在提供查找和寻址信息方面，DNS 和 DHCP 接近于网络控制点。所以它遵循这些服务器的中心化管理的做法，就是同样明智的和有益的㊀。

最普遍采用的网络管理方法是用于网络管理的 FCAPS㊁模型方法。出现了信息技术基础设施库 ITIL®，作为管理企业 IT 基础设施的一个流行指南集。由英国商务部（OGC）开发形成，ITIL 是一个最佳实践框架，它的观点是 IT 组织机构是企业的一个服务提供者。我们将在 FCAPS 模型的上下文内，讨论共同的 IP 地址管理任务，之后在本章末将这些任务的功能映射与 ITIL 过程域相关。

14.1 FCAPS 概述

FCAPS 模型涵盖网络管理实践内的如下关键功能。

（1）F＝故障管理。涉及网络故障的监测和检查，采用诊断、隔离和解决故障等能力。就像网元（例如路由器、服务器和交换机）被监测以便检查故障或中断一样，

㊀ 本章的许多内容映射出参考文献［11］第 6 章（和本章有类似标题）的那些内容。

㊁ FCAPS 是作为管理数据网络的电信管理网络（TMN）框架的组成部分，在 ITU 标准 M. 3400[192] 中定义的。

DHCP 和 DNS 服务应该类似地被监测。也可实现合适的权变措施应对（workaround）机制，例如提供高可用服务的机制。

（2）C＝配置管理。包括网元（含 DHCP 和 DNS 服务器）的准确配置和备份。网元的准确和及时配置，降低了改变管理窗口内的信息准备提供（provisioning）错误数和时间间隔。

（3）A＝记账管理。涉及网络资源使用的跟踪和监督，是就商务配额或顾客权利而言的。涉及访问控制策略的 IP 管理方面、就商务参数的地址利用率以及监测服务水平协议（SLA）符合度等都落在记账管理范围内。

（4）P＝性能管理。处理网元和服务等的性能跟踪以及资源利用率。IP 地址利用率和 DHCP/DNS 服务器性能跟踪，是有效 IP 地址管理的关键要求。

（5）S＝安全管理。包括就网络及其用户的信息安全保障、提供访问控制以及审计日志和安全漏洞检测。IP 地址管理的安全管理包括 IP 地址访问策略、审计、DNS 和 DHCP 安全以及网络上欺诈或违法设备检测。

14.2 共同的 IP 管理任务

使用基本的 FCAPS 功能分类，我们将讨论共同的 IP 管理任务，以“配置”开始，之后逐步移向其他分类。取决于您的 IP 管理系统能力，一些功能可能要求使用多个管理系统。例如，如果您的 IP 管理系统由空白表格程序组成，则您将需要另一个工具来实施故障管理功能。类似地，商用的 IP 管理系统，在其系统内本身就有不同的功能和任务子集合，而其他系统将要求额外的互补性（supplemental）系统。

14.3 配置管理

当多数人想到 IPAM 时，他们主要会认为它是一个配置管理机制。早期的 IPAM 系统事实上仅将焦点放在配置管理上，虽然随着时间推移，许多系统已经扩展到 FCAPS 的其他方面。尽管如此，配置管理仍然是 IPAM 的一个基础功能。在本节，我们将讨论当管理 IP 地址空间和 DHCP/DNS 服务器配置时，所需的共同任务。这些任务与一名 IP 地址规划人员的日常活动有关，涉及 IP 地址、子网、地址空间、域以及 DHCP 和 DNS 配置等其他方面信息的移动、添加和更改。

在 IPAM 上下文内的配置管理包括 DHCP 和 DNS 服务器的配置，这两者分别用于地址租赁和参数指派、名字解析。在最小可能情况下，要涉及 IPAM 有关信息的配置，即地址池和关联参数以及 DNS 配置和区域文件。配置过程也包括高可用部署和服务器等级配置参数的配置，用于基于服务器的或基于仪器的 DHCP/DNS 服务器等。

配置管理功能的结果是，在网络内的每台 DHCP 和 DNS 服务器，将以其文件或参数进行配置，这是它们在网络中实施其相应角色所必要的，如用于一组地址池、故

障切换 DHCP 配置、DNS 区域、参数和选项等的主 DHCP 服务器。从这个角度看，目标是将每台 DHCP 和 DNS 服务器的配置依据其类型（例如 ISC、Microsoft 等）、部署中的角色以及它所服务的网络部分而定。网络部分与指派到每台服务器的一组 DNS 域、子网和地址池的关联有关，并应该与地址空间和域的总体 IPAM 规划一致。

新的、移动的或删除子网的路由器配置，以及要将 DHCP 报文中继到哪些 DHCP 服务器的中继代理信息配置，这些是与 IP 管理密切联系的另一项功能。在市场上几乎没有 IPAM 系统本身就可实施这个层次的路由器集成功能的。从历史角度而言，IP 或服务器团队（team）不同于路由器团队，所以并不鼓励实施团队间实现自动化；毕竟，如果一台路由器被证明是错误配置的，则将汇集到路由器团队，由它们处理。不过一些 IPAM 系统采用本地方式或通过一个 API 调用来支持这个过程的自动化，API 调用可将一个 IPAM 系统子集分配的输出“挂钩”到一个路由器配置工具的输入。类似地，在 IP 清单数据库或表格（spreadsheet）中分配一个子网之后，蛮力（愚蠢的）方法包括将一条电子邮件发送到路由器团队，仅此而已。

14.3.1 地址分配任务

1. 地址块分配

从 IPAM 食物链的顶端开始，考虑实施顶层块分配的任务，使用我们在第 3 章讨论的过程。从顶向下以层次结构方式进行地址空间分配，地址空间分配的规划必须考虑商务需求，这是就以从底向上方式考虑每个应用和每个站点的用户团体这两者的地址容量而言的商务需求。最终，每个站点将从相应的分配得到服务，所以容量规划应该将每个当前及规划的未来站点处的地址需求一起考虑。

如果您没有时间或资源来实施一次完整的容量分析，那么可用于企业组织机构的一条指导规则是，考虑在每个位置的雇员数，并将这个数乘以四。这个数量提供了一个大体估计，并统计了每个雇员的设备以及基础设施设备（如路由器和服务器）。另一方面，如果对于您的组织机构规模而言，您有充足的地址空间，那么您就只需均匀地进行分配即可，就和我们在第 3 章展示说明的那样做就行。

一旦按照站点对地址容量进行了量化估计，则考虑路由拓扑以及如何最佳地对寻址层次结构建模。和 IPAM 全球公司采用的拓扑一样使用一个核心-区域-接入的路由器拓扑，本身就可得到寻址层次结构的一个对应映射。这样一种拓扑的特征是，一个骨干或核心网络连接区域网络，接下来区域网络连接接入网或本地网络。路由器作为拓扑接口，并提供下行（下游，downstream）网络的汇聚。现在将容量数据与拓扑一起考虑，识别在每个层次结构层处总的（roll-up）地址空间。

让我们以例子来形象地说明这个集成过程。IPAM 全球公司的拓扑特征是，一个核心网络服务全球大陆层次。区域层次将北美和欧洲大陆进行细分。17000 名雇员映射到大约 75000 个 IP 地址，此时我们的 10.0.0.0/8 网络有超过 1600 万个 IP 地址，提供了足够的容量，更别提我们的 IPv6 空间了。因此，IPAM 全球公司的 IP 规划人员决定尽可能地采用一种均匀分配策略。最大的配送中心在诺里斯敦，大约有 450 名雇员，最大的昆西分支办事处计划扩展到 200 名雇员。因此，每个配送中心将按照每

种应用[1]接收一个/23 分配（510 个可用 IP 地址），每个分支办事处将按照每种应用接收一个/24（254 个可用地址）。这为每个站点的增长提供了充足的地址空间。

具有最多办事处的北美东部地区，包含 8 个配送中心和 9 个办事处。依据每种应用的相应的地址空间量（rollup）（8 个/23 +9 个/24）约为 1/19[2]。为了针对规模增长而提供充足的地址容量，IPAM 全球公司的 IP 规划人员为每个区域中的每种应用分配了一个/18 网络地址。在定义了这些确定网络地址尺寸的指导原则之后，可实施第 3 章中针对 IPAM 全球公司详细列出的执行过程。

另外可采用一种更集约化的策略，仅为每个站点分配它所需要的容量。这种依据需要的方法，要求比较准确的决策以及随时间跟踪每个站点处的 IP 地址容量需求，从而可保障充足的容量部署。这种方法是更加集约化的方法，但却对地址空间做出了更好的使用，这是较大型组织机构或服务提供商所要求的。对于地址块分配，使用相同的过程，但数学计算和跟踪方面的需求变得有点严格，特别对于不同地址尺寸分配的复合情况更是如此。当为了保留地址空间而对分配进行“适合尺寸”定制（right-sized）时，对先验地监测地址利用率的需求就增加了，其中分配中额外的“空闲（fudge）因子”被控制在一个最小水平。

依据所选择的 DHCP 和 DNS 服务器部署策略以及扩展的需要，您应该就所需每个类型的一个给定尺寸的服务器数量和目标位置，规划出服务器规模（sizing）。依据这个规划，可实施服务器采购、发货/接收，之后是基础层次的服务器配置。因为在过程中的这个点，我们还没有添加任何次级分配，所以这个基础 DHCP/DNS 配置将包括基本的策略定义，以及针对附加地址空间对应于域规划的区域。

对于基础和后续的地址块分配，更新地址规划是必要的第一步骤。但仍有许多工作要做。为了实施该规划，所分配的地址空间应该配置在核心路由器中，以便支持路由表的动态更新。路由器中中继代理信息（目的是中继到 DHCP 服务器（可能是多台））的更新，是另一项必备的任务，但在子网分配任务过程中这确是得到普遍实施的一项任务。附加的后勤任务也许是必要的，目的是在网络接口层次和 DNS 服务层次，将新分配的地址空间添加到服务器访问控制列表（ACL），其中涉及“allow”（允许）选项（例如 allow-query、allow-recursion 等）和视图定义（如果合适的话）。

总之，地址块分配的任务包括如下子任务。

（1）识别要求 IP 空间的站点以及每个站点所需的用户或 IP 设备数量。

（2）确定路由拓扑，这是就地址汇聚需求而言的。

（3）考虑到增长规划，识别出拓扑每个层次的最小分配，并采用分配策略（例如一种均匀的或按需的策略）。

（4）识别 IP 地址清单内的空闲地址空间，并分配所选尺寸的一个地址块。

---

[1] 如果我们向这些站点指派一个不可分的块，则我们将极可能需要总数为雇员数四倍的 IP 地址数量，但相反我们会分配多个类似尺寸的地址块。

[2] 八个/23 = 一个/20，八个/24 = 一个/21。一个/20 + 一个/21 + 一个/24 超过一个/19 的 3/4。

（5）依据需要，设计、采购、安装和配置 DHCP 和 DNS 服务器。

（6）采用所分配网络和中继代理信息，更新路由器配置。

（7）更新 DHCP 和 DNS ACL 配置（如果合适的话）。

（8）管理整体的分配过程，来跟踪位置和在线的服务器。下面讲解每个位置的子网分配。

2. 子网分配

在部署基线地址分配之后，针对子网分配的基础就具备了，子网分配支持用于路由器和主机的个体 IP 地址。商务动机将会驱动子网分配：由于商务扩展需要 IP 地址的新站点，新服务提供的计划（例如 IP 电话（IP 上的语音））以及甚至合并或收购，每种均可严重地影响 IP 地址规划。就将尺寸扩大到期望的容量要求、将容量总和（rollup）映射到支持的路由拓扑以及考虑空闲地址容量和分配策略等方面，可将一种类似的过程用于后续的分配。

子网分配的这项基本任务涉及确定一个可用的子网（该子网将给定位置和应用的地址分配规划求和考虑在内）和 IP 地址规划“数据库”中的子网指派。例如，如果 IPAM 全球公司确定在波兰奥勒冈建立一个新的配送中心，则 IP 规划人员将访问我们的 IP 清单表格，来识别可用的地址空间。在第 3 章给出的北美西部数据空间（10.32.128.0/18）的地址分配分解，如下所示。

北美西部数据 10.32.128.0/18 ***00001010 00100000 10***000000 00000000
旧金山站点 10.32.128.0/23 ***00001010 00100000 1000000***0 00000000
丹佛站点 10.32.130.0/23 ***00001010 00100000 1000001***0 00000000
温哥华站点 10.32.132.0/23 ***00001010 00100000 1000010***0 00000000
菲尼克斯站点 10.32.134.0/23 ***00001010 00100000 1000011***0 00000000
卡尔加里站点 10.32.136.0/24 ***00001010 00100000 10001000*** 00000000
阿尔伯克基站点 10.32.137.0/24 ***00001010 00100000 10001001*** 00000000
盐湖城站点 10.32.138.0/24 ***00001010 00100000 10001010*** 00000000
博尔德站点 10.32.139.0/24 ***00001010 00100000 10001011*** 00000000
埃德蒙顿站点 10.32.140.0/24 ***00001010 00100000 10001100*** 00000000
赛克拉门托站点 10.32.141.0/24 ***00001010 00100000 10001101*** 00000000
阿纳海姆站点 10.32.142.0/24 ***00001010 00100000 10001110*** 00000000
空闲空间 10.32.143.0/24 ***00001010 00100000 10001111*** 00000000
空闲空间 10.32.144.0/20 ***00001010 00100000 1001***0000 00000000
空闲空间 10.32.160.0/19 ***00001010 00100000 101***00000 00000000

如我们在第 3 章讨论的，通过使用一种最佳拟合方法，我们应该考虑使用最小的空闲地址块进行分配。从这个表中，我们看到我们有一个空闲的/24，但对于我们的波兰配送中心（要求一个/23 的分配）而言，这有点太小了。我们继续查看下一个最小的地址块 10.32.144.0/20，并从这个块中分配我们的/23 地址分配。因此，我们将 10.32.144.0/23 分配给波兰配送中心，原始/20 地址块的剩余部分由 10.32.152.0/21、10.32.148.0/22 和 10.32.146.0/23 组成，得到如下所示的内容。

北美西部数据 10.32.128.0/18 ***00001010 00100000 10***000000 00000000
旧金山站点 10.32.128.0/23 ***00001010 00100000 1000000***0 00000000
丹佛站点 10.32.130.0/23 ***00001010 00100000 1000001***0 00000000
温哥华站点 10.32.132.0/23 ***00001010 00100000 1000010***0 00000000
菲尼克斯站点 10.32.134.0/23 ***00001010 00100000 1000011***0 00000000
卡尔加里站点 10.32.136.0/24 ***00001010 00100000 10001000*** 00000000
阿尔伯克基站点 10.32.137.0/24 ***00001010 00100000 10001001*** 00000000
盐湖城站点 10.32.138.0/24 ***00001010 00100000 10001010*** 00000000
博尔德站点 10.32.139.0/24 ***00001010 00100000 10001011*** 00000000
埃德蒙顿站点 10.32.140.0/24 ***00001010 00100000 10001100*** 00000000
赛克拉门托站点 10.32.141.0/24 ***00001010 00100000 10001101*** 00000000
阿纳海姆站点 10.32.142.0/24 ***00001010 00100000 10001110*** 00000000
波兰站点 10.32.144.0/23 ***00001010 00100000 1001000***0 00000000
空闲空间 10.32.143.0/24 ***00001010 00100000 10001111*** 00000000
空闲空间 10.32.146.0/23 ***00001010 00100000 1001001***0 00000000
空闲空间 10.32.148.0/22 ***00001010 00100000 100101***00 00000000
空闲空间 10.32.152.0/21 ***00001010 00100000 10011***000 00000000
空闲空间 10.32.160.0/19 ***00001010 00100000 101***00000 00000000

注意我们的空闲地址块 10.32.143.0/24 现在被指派过的地址块围住或包围。对一个/24 或较小地址块的未来需求可使用这个空间，但其他情况下它是不能使用的。应用一种最佳拟合方法的做法，寻找这些“孤立的”地址块，但如我们看到的情况，它们仍然可能会被放弃使用（即太小而不能使用）。

除了识别并记录所分配的子网外，子网分配过程要求在合适的路由器接口上配置提供子网地址。在子网上的一些单个 IP 地址，需要被指派到基础设施设备，如路由器和服务器。也要求定义和更新 DHCP 服务器配置，这是考虑到地址池（可能有多个）和相应的 DHCP 选项和/或客户端类参数，这些信息是在所分配子网上要求 DHCP（配置）的设备所需要的。

现在和未来在子网上要指派地址的设备，将可能要求 DNS 中的名字解析信息。这个信息以最低限度可施用到域名字到 IP 地址查找的一个转发域以及 IP 地址到名字查找的一个反向域。这要求采用域更新（例如 in-addr.arpa 和 ip6.arpa 域（可能有多个））和名字服务器及静态指派地址的资源记录更新，来定义和更新 DNS 服务器配置。当然，在相应 DNS 服务器上这些域必须存在或必须提供和配置。

取决于您的域拓扑，将一个新的子网添加到一个位置的做法，可能利用一个现有的域，但却不必采取这种方式。一个新域可能需要作为合适 DNS 服务器上的一个子域或一个新的区域加以定义和配置。采用相同方式，也需要添加对应于子网地址的反向域，除非一个较高层 in-addr.arpa 或 ip6.arpa 将驻留（host）相应的 PTR 资源记录时才不需要这么做。

子网分配过程形象地说明了地址分配、指派以及 DHCP 和 DNS 服务器配置任务

间严格的相互关系。取决于您的商务过程，可在地址指派和 DHCP/DNS 配置之前，对子网进行分配或预留。不过，典型情况下，要求实施如下这个完整的步骤集，将一个子网投入运行（生产）。

（1）识别需要子网的拓扑范围内的空闲地址空间。

（2）从合适的地址空间中分配所需尺寸的一个子网，并在 IP 地址规划中记录这个分配。

（3）更新与所分配网络有关的路由器配置。

（4）将针对路由器、服务器或其他子网基础设施设备而分配的地址，人工地指派和配备到位。

（5）如果有必要在子网上服务动态主机，则要设计和配置 DHCP 地址池。这可能要求基于规划使用地址池（可能是多个）的设备要求，将选项、命令（directives）和客户端类进行关联。

（6）定义为子网上服务主机所需的新的 DNS 域，为新的或现有域内的基础设施或设备而定义资源记录，配置合适的 DNS 服务器[㊀]。

（7）通过确认子网的提供和可达性以及验证相应的 DHCP 和 DNS 配置，完成分配过程。

3. IP 地址指派

将 IP 地址指派、去指派和重新指派到个体主机，通常是多数组织机构中最频繁的 IP 管理活动。典型情况下，这与设备（包括路由器、服务器、打印机等）的部署、重新部署或退役相关。就地址指派而言，必须查阅 IP 地址清单数据库，来识别确定一个可用的 IP 地址。如果可能的话，仅为了验证清单的准确性，ping 要被指派的 IP 地址，这种做法将是有用的，但我们将讨论总的清单确保过程，以之作为一项独立的任务。之后，在清单数据库中，要被指派的该 IP 地址应该表示为指派到给定的设备。

实际的物理 IP 地址指派可采用如下方式之一来完成，即人工地（静态地）配置设备、自动配置或使用 DHCP（在这种情形中，我们将假定使用人工的 DHCP，来将所指 IP 地址指派到相应主机）。在静态指派情形中，被指派的地址必须直接配置在设备上，所以除非 IP 地址指派人员也负责物理指派工作，否则这个过程将涉及向设备所有者发送的一封电子邮件或打个电话，其中携带要输入的指派 IP 地址信息。对于自动配置，这个自底向上的指派过程是一个检测问题，而不是自顶向下的指派过程。当使用人工 DHCP 时，为了将设备的硬件地址映射到所指派的 IP 地址，合适 DHCP 服务器（可能是多个）配置文件中的一个表项将是必要的。

多数带有 IP 地址的设备将要求相应的 DNS 资源记录，来支持依据名字的可达性。使用地址指派的 DHCP 方法，DHCP 服务器可配置成，在指派 IP 地址时，更新一台主 DNS 服务器。这种更新将影响用于域名到 IP 地址（A/AAAA）查找的转发域以

㊀ 在一些网络中，要求对 DHCP 地址的资源地址进行预先配置（pre-seeding），目的是在没有实施动态更新的情况下，允许这些地址的用户出现于 DNS 之中（例如为了便利实施 VPN 连接，它要求存在一条 PTR 记录）。

及用于反向（PTR）查找的反向域。如果人工地指派地址，将要求一项类似的DNS更新任务。以这种新的主机信息更新DNS，可能需要在服务器上编辑或更新相应的区域文件或发送动态的更新。

至少在一个企业网络上，您不希望一台自动配置的设备自己来更新DNS，虽然这对于一个团体或ad hoc网络而言可能是合适的。识别存在一台新近自动配置的设备，从而人工地更新DNS，这种做法本身就是一项挑战。如果这样的设备要求DNS中的解析信息，则为了识别IPv6地址，使用一台路由器的日志或子网侦听（snooping）工具就是必要的。

总之，IP地址指派任务包括如下子任务。

（1）确定设备如何得到它的IP地址：通过人工配置、自动配置或通过DHCP。

1）如果是动态DHCP或自动化的DHCP，则确定子网上的当前地址池（如果有地址池的话）是否有容量来支持这台设备；如果有的话，这项任务完成；如果没有，则在DHCP服务器上配置相应DHCP类型的一个地址池和必要的选项参数。

2）如果是人工DHCP，则在设备所处的子网内，识别一个空闲的IP地址，并将该地址指派给该设备，方法是配置DHCP服务器来为设备的MAC地址保留或指派一个人工的DHCP地址。

3）如果是在设备上人工地进行配置，则在设备所处子网内识别一个空闲的IP地址，并将该地址指派给该设备。在设备上人工地配置好所指派的静态IP地址。

4）在所有情形中，以指派的地址更新IP地址规划，不管是一个表格还是其他IPAM工具均可。

（2）确定DNS资源记录是否需要人工地创建和更新。一般来说，对于静态指派的地址，情况就是这样的。对于DHCP指派的设备而言，DHCP服务器可被配置成实施动态更新，虽然在一些情形中，动态更新是不可行的或策略所不允许的，这时就要求相应资源记录的人工更新。

（3）验证地址指派过程的完成，方法是成功地ping该地址，并验证它在DNS中的资源记录。对于通过一个地址池而被指派一个地址的设备而言，可能不需要进行验证；但是，如果需要验证的话，则提前就可能不知道地址。在DHCP租期文件中找到设备的MAC地址，在这种情形中，接下来要做的是，ping相应的地址，以此来确认它的指派。

14.3.2 地址删除任务

如我们所展示说明的，地址分配是一个自顶向下的过程，其中首先是层次结构式地址块的分配，从中可分配子网，从子网可指派IP地址。地址空间的删除，则要求相反的操作，有必要是自底向上的。在下面的地址块、子网和IP、地址被删除之前，就删除一个地址块，将使这些下面的元素处于困境，所以除非您希望出现群众暴乱（mass chaos），否则就要确保采用一个比较可控的过程。

（1）删除IP地址。删除一个IP地址是相对直接的：删除或释放IP清单中的IP地址，如果合适的话就从DHCP中请求M-DHCP项，释放地址租赁，并清除关联的

DNS 资源记录。但是，必须谨慎从事，确保在将该地址指派给另一台设备之前，地址已经由原设备放弃使用，且 DHCP 和 DNS 更新已经完成。例如，简单地在一台 DHCP 服务器上删除一个租赁，并不会强制持有那个租赁的客户端放弃使用那个租赁。设计了 DHCP Force-Renew（强制刷新）消息，强迫一台 DHCP 客户端进入刷新状态，这使一台服务器可能对客户端刷新租赁的尝试回答 NAK，由此释放该地址。但是，强制刷新没有得到广泛实现。

将地址指示为处在“删除进行中”的一个状态或某种类似法，将提醒其他管理员，直到接收到地址可用性的确认之前，不要将那个地址指派给另一台设备。这个确认过程包括：ping 那个地址（也许不断地持续数天时间），并确认在 DNS 和 DHCP 服务器中删除了其关联的数据。

（2）删除子网。当关闭一个站点或合并地址空间时，可能要求删除一个子网。具有要被删除子网上 IP 地址的设备，应该被移除或退役，从而使该子网可用于地址指派（除非，可能是服务子网的路由器情况）。在验证所有 IP 地址都是空闲之后，就将该子网回收到空闲地址空间，用于未来分配。

在释放一个子网时，可能的情况是，将被释放的空间与一个连续的空闲地址块合并，得到一个较大的空闲地址块。沿用我们在上面子网分配小节中的 IPAM 全球公司北美西部数据地址块的例子，如果 IPAM 全球公司决定关闭阿纳海姆分支办事处，则现在就认为它的地址空间 10.32.142.0/24 是空闲的，它与 10.32.143.0/24 地址块是连续的。我们可将这两个地址块合并成单一空闲地址块 10.32.142.0/23。这样做的话，比较清楚的是，这个/23 地址就可指派给一个未来的配送中心（比如）。

（3）删除地址块。地址块删除可能源于退出一个主要商务市场，或站点合并，等等，还有其他原因。一般而言，在宏观层次的地址块可被释放，用于未来指派之前，所有下游 IP 地址、子网、地址池、资源记录和域都应该首先退役。因此，在目标地址块内个体删除 IP 地址任务和子网删除任务完成之后，该地址块本身就可释放了。对于中等到大型分配的任务而言，可能要求项目规划资源，以便验证层次结构上的整体删除。和在删除子网任务中一样，被释放的地址块空间可与连续的空闲空间合并。就像地址块分配一样，就地址池、域、ACL 和资源记录方面，应该考虑与 DHCP 和 DNS ACL 配置有关的附加后勤任务。

14.3.3 地址重新编号或移动任务

对地址块、子网或个体地址的移动或重新编号，这种做法将分配过程与删除过程组合起来。如上所述，分配过程应该是从一种自顶向下的角度实施的，是从低层子网和 IP 地址将被移到的空间进行分配的。随着地址被移到目标的分配空间，删除过程从自顶向下释放地址空间的。本质上而言，必须分配要被移出的地址范围的大小，以便容纳被移出的地址，临时地将与这个设备集合相关联的地址空间翻倍（可能是一种做法）。随着地址被移出，前一个地址空间就可被释放，将地址分配返回到以前的层次。

（1）IP 地址移动。移动一个 IP 地址的做法，可被看做在目的地子网上指派一个

地址，并在当前子网上删除该 IP 地址的组合做法。取决于地址指派的方法和移动类型，可使用不同的战术做法。移动的类型与一台设备物理移动到一个不同的子网（物理移动）及在相同子网或一个不同子网上重新指派 IP 地址（逻辑移动）有关。典型情况下，一台非移动 IP 设备的物理移动将涉及每台 IP 设备的一次“重启”，这要对地址指派过程施加更多的控制。

（2）物理移动。物理移动，意味着断电、移动，之后在目的地位置上对设备上电。对于动态的 DHCP 和自动化的 DHCP 指派设备而言，如果移动的是整个地址池，则应该在一台［相同或不同的］DHCP 服务器上配置目的地地址池。确保服务目的地子网的路由器（可能是多台）被配置成，可将 DHCP 报文中继到配有新地址池的 DHCP 服务器。当这些设备上电时，它们将会尝试刷新它们在旧子网上拥有的最近租赁。确保任何自动化的 DHCP 设备，在上电时发出 DHCPREQUEST，而不仅仅继续使用它们的旧 IP 租赁；如果这些设备假定旧［无穷期限］租赁是有效的，则将要求人工干预，重置该设备的地址。否则，DHCP 服务器将 NAK（否定应答）每个客户端的 DHCPREQUEST 尝试。客户端们将返回到 Init-Reboot（初始化重启）状态，并发出一条 DHCPDISCOVER 报文，来得到一个新的地址租赁。DHCP 服务器以新的目的地址池内一个新的地址租赁作为应答。对于 DHCPv6 客户端，可使用一个类似的过程。一旦所有设备都以物理方式移动，则服务旧子网的地址池可做退役处理。

一台 M-DHCP 设备的物理移动，包括在服务新子网的 DHCP 服务器中创建 M-DHCP 表项，并在以前的 DHCP 服务器上删除相应表项。如果使用的是同一台 DHCP 服务器，则简单地编辑与设备的 MAC 地址关联的 IP 地址。当设备在新子网上上电时，它应该遵循动态 DHCP 和自动化 DHCP 类似的一个过程，进行一次 DHCPREQUEST 尝试，如果 DHCP 服务器应答为 NAK，则接下来是（返回到）发出一条 DHCPDISCOVER，使用标准的 DHCP 过程进行地址重新指派。

移动自动配置其 IPv6 地址的一台设备的做法，将会使设备通过路由器发现，检测到它的新子网，还有对应的子网策略（包括 DHCPv6 服务的可用性）。如果进行地址的自动配置，那么该设备就进行自动配置，之后通过重复地址检测来验证地址的唯一性。如果使用 DHCPv6，则接下来是正常的 DHCPv6 过程，来得到一个 IPv6 地址及关联的参数。在一些情形中（当在路由器通告中设置 O 比特时），可同时使用自动配置和 DHCPv6。

可由 DHCP 服务器来实施 DNS 资源记录更新，或如果这些 DHCP 案例是被禁止动态更新的话，则人工更新。

人工配置设备的物理移动的做法，要求从 IP 清单中指派一个 IP 地址，并当该设备在新子网上上电时，将新的 IP 地址人工配置在该设备中。在此时，旧地址就可释放了，虽然在验证地址可用性之前为防止相应地址的未成功重新指派，一个中间“删除进行中”状态可能是有用的。为了反映设备的新 IP 地址，也应该更新 DNS 资源记录。

在所有这些情形中，应该使用 IP 清单来识别目的地子网或地址池上的空闲地址（可能是多个），并当确认设备移动之后，在旧子网上释放地址和相应的 DNS 资源

记录。

(3) 逻辑移动。逻辑移动有点挑战性，原因是逻辑移动不必涉及一台设备的重新初始化。对于DHCP设备，应该在［相同或不同］的DHCP服务器上配置包含目的地IP地址的一个地址池。在移动日期之前，地址池或设备的租赁时间应该被逐步减少。例如，如果一个正常租赁时间是1个星期，那么在要移动的那个星期，它应该小到1天（比如），在移动的那天，应该小到2-6小时。恰在您将租赁时间更改为天之前，一台设备可能刷新得到一个星期那么长的地址租赁，所以直到那个星期一半过去之前（或依据您的T1时间选项设置），它不会尝试进行租赁刷新。因此如果您的正常租赁时间是2个星期，则在计划的地址移动之前，逐步将2个星期的租赁时间减少。在移动的那天，如果所有设备都要在几乎相同时间移动，这样做比较重要的话，那么将租赁时间设为一个最小㊀时间。如果移动一致性不是至关重要的话，那么保持在数小时量级上的租赁时间，应该可得到在数小时内的一次完全移动任务。

在这个场景中，建议，如果可以采用地址改变比较紧密地映射DNS信息更新的话，则由DHCP服务器实施DNS更新。A-DHCP设备的人工干预会是必要的，除非它们确实遵循租赁刷新策略，而不拥有无穷的租赁时间。

人工编址设备的移动，遵循物理移动中的相同过程。从IP清单中指派一个目的地IP地址，并将新的IP地址配置在设备上。一旦确认，就可释放旧地址，也应该更新DNS资源记录，以便反映设备的新IP地址。

一台自动配置设备的逻辑移动，可以如下方法实施，通过在服务相应子网的路由器上，在邻居（路由器）发现过程中，将其通告的首选地址寿命和有效地址寿命值逐步降低。缩短地址前缀（设备要从该前缀移走）的这些定时器值，同时引入带有一个"正常"地址寿命的新前缀，这种做法将使自动配置的设备自动地实施这种逻辑移动。一旦所有设备都移动了，且前一前缀的有效寿命超期，则可清除该前缀。

(4) 子网移动。移动一个子网，可能涉及两个结果中的一个结果：将子网及其被指派的IP地址都移动到另一个路由器接口，保留当前地址指派；或移动到另一个路由器接口，但要求新的子网地址。我们将子网重新编址任务与后一种情况放在一起，原因是子网重新编址也会导致一个新的子网地址，虽然并不必将该子网移到另一个路由器接口上。前一种情况要求考虑在层次结构内地址空间回收（rollup，汇总），但一般包括修改并验证路由器地址提供的与规划的要一致，同时在必要情况下，更新路由表和DHCP中继地址。

由于一次物理移动或较高层次的重新编址导致的一个子网移动，会要求更多一些的工作。一次物理移动（其中设备要被物理地移动（例如当一个办事处搬迁时））本质上是中断性的。可在目的路由器接口上分配和准备目的地子网，还有上面描述的其他任务，它们与保留静态地址以及更新DHCP和DNS配置有关。当每台移动的设备插入时，它将需要使用新的地址进行人工重新编址，和/或得到与子网有关地址池上

㊀ 取决于网络流量和服务器性能考虑因素，最小时间可在数分钟或数小时的量级上。租赁时间越短，则将发送的DHCP报文就越多，但DHCP客户端移动可被编排的时间就越准。

的一个 DHCP 租赁，过程见上面对 IP 地址移动所描述的情形。类似地，逻辑地址移动或重新编址，遵循每台设备逻辑 IP 地址移动的过程。

在所有设备都从旧子网移到新子网后，旧子网就可遵循删除子网过程进行释放。

（5）地址块移动。将带有低层子网和 IP 地址的宏层次地址块，要求仔细的项目规划和实施。目的地地址块的分配应该遵循针对地址块分配列出的那些任务。假定一次移动仅是重新编址，应该分配一个类似尺寸的目的地址块。如果移动是由地址合并或扩展导致的，或可能产生这样的机会，则目的地址块分配的大小应该依据低层容量需求和拓扑架构，见地址块分配一节讨论的情形。一旦完成分配，就可开始亚层次的分配和子网分配。之后 IP 地址和地址池的移动，遵循针对 IP 地址移动所描述的过程。随着 IP 地址和子网完全从其旧指派中移出，且移动被确认、其对应的资源记录被清除时，这些地址和子网就可退役或释放。

14.3.4　地址块/子网分割

将一个地址块分割，涉及从一个给定源地址块中产生两个或多个较小尺寸的地址块。为了释放地址空间或甚至作为地址空间亚层次分配的一种方式，分割就是必要的。在前一种情形中，在一个子网内的地址可被合并到该子网的前一半，并释放在后一半中的指派。在这种场景中，将地址块分割的做法，就得到一个被占用的子网（前一半）和一个空闲的子网（后一半）。一些组织机构历史上分配了区域性的地址块，那么可将这些地址块分割，在地址层次结构中指派较低层次的亚地址块和子网。在某种意义上而言，这是地址块分配的一种形式。

注意，一般而言，将一个地址块分成两个地址块，即当有一个网络和一个广播地址的以前单一网络现在变为两个网络，每个新网络有一个网络和一个广播地址时，将有可能使两个以前可用的地址成为不可用的。例如，192.168.24.0/24 网络有网络地址 192.168.24.0 和广播地址 192.168.24.255。将这个地址块分成两个/25 网络 192.168.24.0/25 和 192.168.24.128/25，则使以前可用的地址 192.168.24.127 作为新的地一个网络的广播地址、192.168.24.128 作为第二个网络的网络地址。

当分割地址块时，要注意 DNS 反向区域的影响。如果所得到的两个子网的 DNS 权威机构仍然在一组管理员控制之下，则原始的 in-addr.arpa 或 ip6.arpa 区域也许不需要改动。但是，如果一个得到的分割地址块或子网，其设备在由一个独立的委派权威控制下的 DNS 中得到管理，那么原始的反向区域也要求分割。这涉及产生对应于所得到分割子网的两个反向区域，并通知负有职责的分割子网的父反向区域管理员，使之正确地将反向区域树下的权限委派给负责权威信息的正确的 DNS 服务器集。

将一个地址块分割的做法，不必仅约束为分割成两半（比如一个/24 网络分割成两个/25 网络）。一个分割可被用作从一个/20 网络中切出一个/23 网络，如在子网分配一节中将地址空间指派给我们新的波特兰配送中心时，我们所做的那样。这个分割得到一个/23 网络（我们将之指派给波特兰）和空闲空间（由一个/23、一个/22 和一个/21 网络组成）。在这个例子中，我们遵循我们的最优分配策略，保留大型的地址块。另外，我们可简单地将我们的/20 网络分成八个/23 网络，除非策略使用等尺

寸的分配，否则这种方法是严重浪费的，这种策略是均匀分配策略，它与按需分配策略是相反的。

总之，分割一个地址块的过程，类似于分配一个地址块的过程。要被分割的地址块被不断地划分，直到得到期望的地址块尺寸时才不再划分。剩余的空闲地址块保持原状，或也被分割为期望地址块大小的相同尺寸，从而得到一个均匀的地址块分割。DNS 对反向域树和管理委派的蕴含意义是必须要考虑的。且要牢记在心的是，从分割得到的每个网络，都会产生一个附加的网络和广播地址。

14.3.5　地址块/子网合并

一次合并将两个连续的等尺寸地址块或子网组合成单一地址块或子网。在地址块删除一节我们看到了合并地址块的一个例子。在释放阿纳海姆地址块 10.32.142.0/24 之后，我们将其合并到一个连续的空闲地址块 10.32.143.0/24，得到单一地址块 10.32.142.0/23。可实施后续的合并，从而合并连续地址空间的较小地址块。合并仅对相同尺寸的连续地址块是有效的。将一个/25 网络和一个/24 网络合并是无效的，原因是没有包括在合并中的“另一个/25 网络”一定还保持在唯一识别使用状态。但是，两个连续的/25 网络和一个邻居/24 网络可被合并，形成一个/23 网络。这两个/25 网络将首先被合并形成一个/24 网络；之后这个网络与另一个/24 网络可被合并，产生一个/23 网络。

被合并地址块的渐次增加的做法，也会要求 DNS 反向区域的更新，以便将低层设备资源记录合并到一个“合并的”反向区域，以此反映得到的合并子网。

14.3.6　DHCP 服务器配置

如我们在本书第Ⅱ部分讲到的，DHCP 服务器配置是一项关键的 IPAM 任务。如我们已经讨论的，迄今为止讲到的地址管理任务，对 DHCP 服务器配置具有重大影响。DHCP 服务器配置不仅仅是地址池创建、移动和删除，虽然其他功能的范围会受到 DHCP 服务器厂商能力的约束，但也不仅仅如此这些功能（还有其他功能）。DHCP 服务器配置的主要参数是

（1）DHCP 地址池。地址范围以及关联的 DHCP 选项，还有用于动态客户端、自动化的客户端和人工配置客户端的服务器策略。

（2）客户端类。匹配值的参数（例如 vendor-class-identifier = ’Avaya’ 4600）及关联的 allow/deny 池，还有 DHCP 选项和服务器策略。

（3）针对主/故障切换或分割范围的高可用性参数设置。

（4）服务器活动的配置，例如动态 DNS 更新以及其他服务器命令（directives）和参数。

实际的服务器配置语法和界面将取决于服务器类型。例如，ISC DHCP 服务器可通过编辑 dhcp.conf 文件进行配置，而微软 DHCP 则要使用一个 Windows MMC 界面进行更新。这两个厂商和其他 DHCP 厂商也提供命令行界面或 API 来实施配置更新。在第 7 章讲到的 DHCP 部署，在每台服务器的配置中也扮演了一定的角色。要了解这些

产品和其他产品，请参考所购买软件厂商的文档材料。

（1）地址指派/DHCP 和 IP 地址管理。为了确保唯一性，需要记录各项静态 IP 地址指派。在所分配的子网内，无论是静态指派的还是动态指派的，都应该跟踪 DHCP 地址池，以便提供子网内地址指派的总体视图。虽然在一个表格内跟踪个体 DHCP 租赁是不容易实施的，但至少应该实施表格或数据库内地址池分配的跟踪记录。这将有助于确保随时间消逝，保持唯一一致的地址指派。

这个合并的地址指派数据库提供了所知层次的 IP 地址清单。IPAM 全球公司的团队为静态设备（例如路由器、交换机和服务器）在每个子网上都指派了一个一致的 IP 地址集。该团队也为打印机定义了许多人工配置的 DHCP 地址，为 DHCP 客户端设备（如笔记本计算机和 VoIP 电话）间共享定义了地址池。针对清单个体地址和地址池指派，我们为每个站点创建了一个新的标签（tab）页（这个表格正变得非常巨大）。对于某些设备，附加的“备注”信息对于跟踪也是有用的，例如厂商联系方式、支持信息，资产信息等。

除了跟踪 IP 地址指派外，必须实施相应 DHCP 服务器（可能是多台）的配置，以便支持 DHCP 客户端得到地址。DHCP 服务器的配置，涉及采用对应于地址规划内那些指派的地址范围，配置服务器。在图 14-1 中，我们分配了地址 10.16.128.50 ~ 10.16.129.240 作为一个 DHCP 池，所以必须在一台 DHCP 服务器上定义这个地址范围。另外，为了正确地配置不同类型的客户端，在 DHCP 服务器上需要配置客户端类信息、选项和其他配置参数。这项配置操作可采用如下方式实施，针对 ISC 的 DHCP 服务器使用一个文本编辑器，对于微软 DHCP 服务器使用微软管理控制台（MMC），或使用一个 IPAM 工具，它支持针对部署于您所在网络中 DHCP 服务器类型，实施自动化的 DHCP 服务器配置。使用一个 IPAM 工具的主要优势是，IP 清单信息容易支持定义 DHCP 池，且可在 IPAM 系统中定义许多 DHCP 服务器配置信息，之后将这些配置施用到多台 DHCP 服务器，而不是在多台服务器上重复性地进行定义。

14.3.7 DNS 服务器配置

按照本书第Ⅲ部分的描述，像 DHCP 一样，DNS 服务器配置是一项至关重要的 IPAM 功能。如我们所看到的，DNS 配置与地址分配、指派、移动和删除密切地联系在一起。前面讨论的这些任务影响 DNS 域、资源记录，并可能影响服务器配置参数。关键的 DNS 服务器配置参数如下。

（1）域。在 DNS 服务器上添加、修改或删除域/区域。

（2）资源记录。添加、修改或删除资源记录。

（3）服务器、视图和区域配置。设置和修改选项参数会影响 ACL、服务器配置等。

实际的 DNS 服务器配置语法将取决于服务器类型。可通过编辑服务器上的 named.conf 和关联的区域文件，配置 ISC BIND 服务器。支持 DDNS 的 DNS 服务器也支持采用这种方式的资源记录更新。使用 nsupdate 或类似的 DDNS 机制的做法，提供了实施增量式更新而不需要人工编辑区域文本文件并重载相应区域（例如使用 rndc）

位置	地址块/子网	IP地址	地址和设备类型	备注
旧金山	10.16.128.0/23	10.16.128.1	静态——路由器	SanFran VoIP 子网路由器 1
		10.16.128.2	静态——路由器	SanFran VoIP 子网路由器 2
		10.16.128.3	静态——路由器	SanFran VoIP 子网路由器 HSRP 地址
		10.16.128.4	静态——DNS 服务器	技术支持联系 Fred Jones
		10.16.128.5	静态——FTP 服务器	
		10.16.128.6	静态——文件服务器	SanFran 备份
		10.16.128.7	静态——文件服务器	西雅图的备份
		10.16.128.8	静态——文件服务器	菲尼克斯的备份
		10.16.128.9		为以后发展保留
		10.16.128.10	静态——IPPBX	IP PBX-SF1
		10.16.128.11	静态——IPPBX	IP PBX-SF2
		…		
		10.16.128.20	工程实验室服务器	联系工程部寻求帮助
		10.16.128.21	工程实验室服务器	联系工程部寻求帮助
		10.16.128.22	工程实验室服务器	联系工程部寻求帮助
		…		
		10.16.128.50 ~ 10.16.129.240	VoIP DHCP	技术支持联系 Mary Smith
		…		

图 14-1 IP地址的样例清单表

的方法。DDNS更新仅适用于资源记录增加/改变/删除，所以任何区域或服务器配置参数改变或区域添加或删除，都仍然需要进行文本文件编辑，并重新载入named.conf和/或被影响的区域。如我们在第11章中讨论的，您的DNS服务器的部署模型也在服务器配置中扮演了一个角色。

（1）DNS和IP地址管理。给定IP地址与反向域之间的直接关系，主机名和其他主机信息条件下，清楚的是，DNS是IP地址管理的一个关键组件。在一个IP子网上的各主机被指派主机名（目的是方便人们理解）和IP地址（目的是支持通过IP报文的通信）。DNS提供主机名和IP地址之间至关重要的联系，这使IP应用比较容易使用。

从一个IP地址管理角度看，明显的是，反向DNS域与IP地址块和子网分配有直接关联。这些域是直接从它们相应的IP地址推算得到的。IPAM全球公司是从其ISP或域注册处得到ipamworldwide.com域名的。在这样做时，IPAM全球公司提供三个DNS服务器地址，可将查找ipamworldwide.com后缀解析的迭代查询，定向到这三个地址。在指派万维网、电子邮件和有关的面向因特网的服务器后，这个域后缀可帮助IPAM全球公司创建一个全球因特网（点）。

在组织机构内，这个域名也用在内部网上。为公司、销售、工程和物流团队，定义了子域。工程子域（eng.ipamworldwide.com）被委派给工程团队的DNS管理员，

而其他子域将在 IT 组内采取中心式管理。可在不影响 IT 团队管理工作的条件下，工程团队可进一步在 eng. ipamworldwide. com 之下创建子域。通过委派 eng 子域，IT 团队赋予工程团队管理所有 eng 子域主机及其子域的权力。

在遵守中心式 IP 地址清单的常规思维过程中，得到的结论是，应该实施与每个 IP 地址关联的主机名和资源记录跟踪。在我们的 IP 清单表（我们刚刚回顾了 IPAM 全球公司旧金山办事处的情况）的基础上，我们可针对个体设备跟踪这个信息，方法是在我们的表上简单地插入 FQDN 列，如图 14-2 所示。

地址块/子网	IP 地址	地址类型	FQDN	备注
10. 16. 128. 0/23	10. 16. 128. 1	静态——路由器	router-sf01. ipamworldwide. com.	SanFran VoIP 子网路由器 1
	10. 16. 128. 2	静态——路由器	router-sf10. ipamworldwide. com.	SanFran VoIP 子网路由器 2
	10. 16. 128. 3	静态——路由器	router-sf11. ipamworldwide. com.	SanFran VoIP 子网路由器 HSRP 地址
	10. 16. 128. 4	静态——DNS 服务器	ns-sf01. ipamworldwide. com	技术支持联系 Fred Jones
	10. 16. 128. 5	静态——FTP 服务器	ftp-sf. ipamworldwide. com.	
	10. 16. 128. 6	静态——文件服务器	filecab-sf. ipamworldwide. com.	San Fran 从属
	10. 16. 128. 7	静态——文件服务器	file-dr. ipamworldwide. com.	西雅图备份
	10. 16. 128. 8	静态——文件服务器	file-phx. ipamworldwide. com	菲尼克斯备份
	10. 16. 128. 9			为发展增长预留
	10. 16. 128. 10	静态——IP PBX	denalo1. corp. ipamworldwide. com.	IP PBX-SF1
	10. 16. 128. 11	静态——IP PBX	denalo2. corp. ipamworldwide. com.	IP PBX-SF2
	…			
	10. 16. 128. 20	工程实验室服务器	eng-sf1. eng. ipamworldwide. com.	联系工程部寻求帮助
	10. 16. 128. 21	工程实验室服务器	eng-sf2. eng. ipamworldwide. com.	联系工程部寻求帮助
	10. 16. 128. 22	工程实验室服务器	eng-sf3. eng. ipamworldwide. com.	联系工程部寻求帮助
	…			
	10. 16. 128. 50 ~ 10. 16. 129. 240	用于 VoIP 电话的 DHCP 池		技术支持联系 Mary Smith
	…			

图 14-2 带有 FQDN 的样例清单表

在上例中，我们仅跟踪每台静态确定主机的 FQDN。从 DHCP 地址池得到租赁的各主机，是通过动态 DNS 更新它们在 DNS 中的主机名信息的。我们需要确保我们正确地将这个清单信息转录到 DNS 服务器配置之中。从这个“数据库”，我们可推算得到对应于每台主机的 A、AAAA 和 PTR 记录。我们可在表格上扩展列，来跟踪与给定主机关联的其他资源记录，例如 CNAME、MX 等。

14.3.8 服务器升级管理

DHCP 和 DNS 服务器软件的新版本是周期性发布的，目的是解决安全弱点，提供缺陷修正，或提供新的功能。实施一次升级的紧迫性，通常取决于要解决什么，安全弱点当然是最高紧迫性的。典型情况下，升级过程是特定于厂商的，并可能要求匹配硬件平台和操作系统，只有在这种匹配条件下，升级的版本才被证明是可运行的。人们希望的是，底层操作系统需求将仅对新的功能特征引入时才会发生改变，是不针对安全或缺陷修正的，但这是由厂商策略所控制的。

在一个整体性的升级包中，大量厂商 DHCP/DNS 工具（appliance）升级批量进入操作系统升级才是必要的。因为典型情况下，工具厂商和硬件平台一起提供操作系统，对于其 DHCP 和 DNS 服务的较新版本，如有必要，他们应该发布兼容性升级。多数工具解决方案支持升级软件包的中心化阶段式进行，是要部署到分布式工具的，这极大地简化了一个基于软件的升级过程难度。

如果您正在您自己的硬件上运行 ISC、微软或其他厂商的 DHCP 或 DNS 守护进程，则您将不仅需要不断了解 DHCP/DNS 安全更新，而且需要了解运行在硬件上影响相应操作系统的那些更新。

14.4 故障管理

故障管理不仅包括故障检测，而且包括告警通知、故障隔离能力、故障跟踪和问题解决过程。针对故障和事件，对 DHCP 和 DNS 服务器实施监测，支持最小化服务中断的一种先验方法。在一个良好设计的网络服务架构中，不用考虑个体 DHCP 或 DNS 服务器中断的情况，客户端应该能够得到地址租赁，并解析域名。不过，对这种一次中断的检测是重要的，原因是（服务）中断会减少客户端可用来得到这些服务的服务器数量，因此在出现一次额外的服务器故障情况下，会增加对服务中断的脆弱性。例如，在一个 DHCP 故障切换部署中，一台服务器的故障将仅留下一台服务器可用于服务 DHCP 客户端。在这样一个场景中，失效服务器的检测，可有利于及时地（虽然不是恐慌的）解决服务器中断。

14.4.1 故障检测

取决于所部署 DHCP 和 DNS 服务器支持的能力，可使用各种方法来实施故障检测。这些方法范围广泛，从专有的轮询或通知，到 syslog 扫描和/或转发，到基于 SNMP 的网络管理系统的 SNMP 轮询和陷阱检测。另外，一些商用 IP 管理系统提供系

统内或专用的监测方法，特别对于基于仪器设备的产品更是如此。因为仪器设备是完全自包含的解决方案，所以不仅集成了 DHCP 和 DNS 服务，而且集成了一个硬件平台和操作系统，该厂商具有如下能力，即在硬件、操作系统和 DHCP/DNS 层次，可完全访问与仪器设备有关的故障信息。

除了监测 DHCP 和 DNS 服务器的状态外，正如可由服务器报告的，监测处于暂停的服务，是一种好的想法。在如下情况下会出现服务暂停，即如果一项服务正在运行，但处于这样的一个状态，其中在提供地址租赁或解析 DNS 查询方面，它不能实施它的角色任务。可采用如下步骤检测服务暂停，即分析所接收和处理的对地址租赁或查询事务的后续轮询，验证大于零的差异计数（differential counts），其中假定在一天的那个特定时间处正常事务率是非零的。

服务测试的另一种应需形式涉及向服务器发送一条 DNS 查询或 DHCPDISCOVER（或 SOLICIT）报文，并验证接收到一条正确的响应报文。这种策略提供了某种保障，即服务不仅是运行的，而且可对客户端做出响应。基线是某种形式的服务功能性故障检测，对于一名端用户可能认为是一次故障或中断的情形，提供一个较真实的映射关系。

除了监测 DHCP 和 DNS 服务器外，对 IP 管理系统本身的监测，为确保可访问 IP 地址以及 DHCP 和 DNS 服务器配置信息（在中断妨碍得到这些信息的其他情况下可使用这种方法），提供了方便。采取最小模式（最低限度），数据存储的备份或分布可提供一个快照，出现一个站点中断或灾难时，可重建信息。

联网设备和通信链路的监测，是通用网络监测的一项共同实践，它可对影响客户端到达 DHCP 或 DNS 服务器的网络中断提供深入全面的信息。在对一个特定问题排查或故障排查过程中，这项额外的信息是非常有帮助的。

故障相关是对从多个网元或管理系统除所接收个体故障的分析，有助于隔离故障集合的根源。例如，来自一台层 2 交换机、一台路由器和一台 WAN 接入设备的各故障可被整体性地分析，揭示出这三种故障是相关的，可能的根源是一次链路中断。故障相关是大型网络管理系统的一项共同功能，且如果您的 IP 管理系统提供报警反馈，就能够将之馈入到一个较高层的报警关联功能。不论故障相关是由一个网络管理系统自动实施的，还是人工地比较来自多个系统的信息，这个过程都暴露了用于故障分析的一个较广的数据集合，目标是将一个故障隔离到一台给定服务器、一条链路或一个网元。

对于负责管理一个 IP 网络的那些人员来说，故障管理能力是一项重要考虑，且关键的 DHCP 或 DNS 网络服务应该处于被监测的那些网元行列之中。可取得故障影响的缓解效果，方法是部署高可用的配置，以便最小化任意个体组件一次故障导致的端用户影响。

14.4.2　排错和故障解决

（1）IP 地址排错。各种工具可用于排查 IP 指派、DNS 和 DHCP 故障，其中一些甚至可由您的 IPAM 厂商提供。为了验证或识别 IP 地址指派，有意进行的或无意注意

到的，各种发现技术将被证明是有益处的。从一种简单的 ICMP Echo 请求、ping、traceroute、nmap 或 SNMP，可用各种工具，尝试联系个体主机或查看路由器或交换机 ARP 表。许多 IPAM 系统至少继承了一种形式的发现技术，以便提供 IP 地址指派的验证或协助排错。

(2) DNS 排错。除了服务器可达性和服务器/服务状态检查外，对 DNS 解析的排错，是诊断和解决 DNS 问题所需要的一项关键功能。ISC 提供了一对配置检查工具，作为载入网络配置或区域文件之前的一次语法检查，这种工具是有用的。

配置文件检查。named-checkconf (144) 命令实施 named. conf 文件的语法检查。这个命令的语法是

named-checkconf [-v] [-j] [-t directory] [filepath] [-z]

命令参数定义如下。

1) -v：打印 named-checkconf 的版本。

2) -j：当载入一个区域文件时，如果存在日志文件，则读入该文件。

3) -t directory：为了处理所包括的命令 (directive)，改变 directory 的 root (chroot)。

4) filepath：named. conf 文件的路径，默认为/etc/named. conf。

5) -z：按照 named. conf 中定义的，载入主区域文件，来验证正确的载入。

区域文件检查。named-checkzone (144) 工具提供对一个特定区域文件的语法检查。这个命令的语法为

named-checkzone [-v] [-j] [-d] [-q] [-c class] [-k mode] [-n mode] [-o filename]

[-t directory] [-w directory] [-D] [zonename] [filepath]

命令参数定义如下。

1) -v：打印 named-checkzone 的版本。

2) -j：当载入一个区域文件时，如果存在日志文件，则读入该文件。

3) -d：激活调试。

4) -q：静默模式，1 = 错误，0 = 没有错误。

5) -c class：指定区域类：默认为 IN。

6) -k mode：以 fail (失效)、warn (警告) (默认的) 或 ignore (忽略) 三种模式 (mode) 之一对主机名实施 check-name 检查。

7) -n mode：以 fail (失效)、warn (警告) (默认的) 或 ignore (忽略) 三种模式 (mode) 之一检查 NS 记录是否不正确地将 IP 地址用作 RData。

8) -o filename：将区域输出写到 filename。

9) -t directory：为了处理包括命令，将 root (chroot) 更改到 directory。

10) -w directory：为了处理包括命令，将当前工作目录更改到 directory。

11) zonename：正被检查的区域的域名。

12) filepath：到区域文件的路径。

名字服务器检查。在最流行的 DNS 诊断工具中，有两个是 nslookup (名字服务器查找) 和 dig (域名信息探索器)。nslookup 被包括在 Windows DNS 安装中，在 BIND

软件发布中都带有 nslookup 和 dig。nslookup（145）是一个简单的工具，它支持对一台 DNS 服务器的查询。如今，多数管理员首选 dig，该工具提供对查询构造形成、解析器配置覆盖（override）、输出格式化等的更多细节和控制。为了使用 nslookup 而实施单一查找，简单输入

nslookup lookup-value [name server]

其中 lookup-value 是要查找的主机域名或 IP 地址，name server 是要查询的服务器名或 IP 地址。在 lookup-value 之前，可能包括如下面指定的其他选项，每个选项名都有一条横线作为前缀（例如-timeout = 5）。nslookup 的交互模式可如下触发，输入不带参数的 nslookup，或输入 nslookup，后跟一条横线、空格和名字服务器主机名或 IP 地址，如

nslookup - 172.18.71.105

交互式模式支持输入命令，形成查询并执行查询。可使用如下交互式模式命令。

1）host [nameserver]（主机 [名字服务器]）：在指定 nameserver 或默认服务器上查找 host。这类似于上面描述的命令行格式。

2）server domain：使用当前 [默认的] 服务器查找 domain，将默认服务器更改为 domain 的权威服务器或 domain 字段中指定的 IP 地址。

3）lserver domain：使用当前 [默认的] 服务器查找 domain，将当前服务器更改为 domain 的权威服务器或 domain 字段中指定的 IP 地址。

4）exit（退出）：退出交互式模式。

5）set option [= value]：为影响查找行为，设置选项；通过在 nslookup 命令行上的选项名前面带有一条横线，也可在非交互式模式中使用如下选项。

① all：显示当前选项值和当前 [默认] 服务器和主机。

② class = value：在查询内设置 Qclass；有效值是 IN、CH、HS 或 ANY。

③ [no] debug：debug 激活所有响应报文的显示，nodebug 则去活这种显示。

④ [no] d2：d2 打开调试，nod2 关闭调试显示。

⑤ domain = name：将域搜索列表设定为域 name。

⑥ [no] search：search 配置使用域搜索解析器配置，将这样的域附加到至少有一个点（dot）的不完全合格的搜索之后。nosearch 去活使用这个搜索列表。

⑦ port = value：将 TCP/UDP 端口号设置为 value；默认为 53。

⑧ querytype = value：将 Qtype 设置为到要查询的一个资源记录类型。

⑨ type = value：与 querytype 相同。

⑩ [no] recurse：recurse 发出一条递归查询，norecurse 不发出这样的查询。

⑪ retry = number：设置查询重试的次数（number）。

⑫ timeout = seconds：设置等待一条应答的秒数（seconds）。

⑬ [no] vc：vc 指令 nslookup 使用 TCP，novc 使用 UDP。

⑭ [no] fail：如果接收到一个 SERVFAIL 或一个转荐（referral），则 nofail 将 nslookup 设置为尝试下一台名字服务器；fail 不尝试下一台服务器。

域信息探索器。dig[146] 支持使用标准 DNS 消息形成一条 DNS 查询，它模拟一个解

析器或递归服务器。dig 提供了可被发送到一台 DNS 服务器的一条查询的格式的粒度控制，目的是分析得到的响应。

dig 命令的一个常见范例用途，简单地请求对一个主机名的解析：

dig @ ns1. ipamworldwide. com A ftp-sf. ipamworldwide. com

这个范例将导致对 ftp-sf. ipamworldwide. com 的一条 A 记录查询发送到 DNS 服务器 ns1. ipamworldwide. com。dig 工具有效的可能参数包括以下内容。

1）@ server：其中 server 是 DNS 服务器的域名或 IP 地址，要向该服务器发出查询。如果没有指定这个参数，则 dig 将查询在客户端的解析器配置中列出的 DNS 服务器。

2）－b address：将查询的源 IP 地址设定为 address。对于测试 ACL 或视图而言，这是有用的。

3）－c class：要查询的 DNS 资源记录的类；默认为因特网（Internet）。

4）－f filename：支持发出连续查询，按照在指定 filename 中列出的情况，即一条查询一行。就像您让一条 dig 命令格式化给定查询一样，对文件的每行进行格式化（不需要在每行都指定“dig”）。这有利于在一步中自动化地测试一组关键性的解析。不要忘记老板心爱的解析。

5）－k filename：对查询签名，并使用 TSIG 验证响应签名，TSIG 是在 filename 中指定的事务签名。TSIG 密钥必须与 DNS 服务器的 named. conf 配置中定义的密钥匹配。

6）－p port：指定用于查询的目的地 UDP（或 TCP）端口；如果没有指定，则使用默认 DNS 端口 53。

7）－q name：显式地识别在查询中要用的属主 name，而不使用“bare”（直接的、裸的）name 参数。即 dig －q sf-ftp1 = dig sf-ftp1。

8）－t type：显式地识别要使用的查询类型，而不使用“bare”（直接的、裸的）type 参数。除非指定-x，否则默认类型是“A”，指明一次 PTR 查找。

9）－x address：为指定的 address，确定一个 PTR 查找。这个选项支持输入一个 IPv4 地址或 IPv6 地址，但如果和类型 PTR 使用-t，则名字必须格式化为一个 . arpa. 名字。

10）－y［hmac：］name：key：指定一个显式的 TSIG 密钥（而不是使用-k 选项引用一个文件）。hmac 字段指明密钥算法，默认为 HMAC-MD5，name 字段是 TSIG 密钥名，key 是密钥本身。当使用-y 选项时，应该小心谨慎，原因是可从输出或命令外壳历史中看到密钥。TSIG 密钥必须匹配 DNS 服务器的 named. conf 配置中定义的密钥。

11）－4：使用 IPv4 传输发送查询。

12）－6：使用 IPv6 传输发送查询。

13）name：要查询的属主名。dig 支持国际化的域名（IDN），所以可指定非 ASCII 名字。

14）type：在查询中要查询的查询类型。默认类型为 A。

15）class：要查询的资源记录类。默认类为因特网。

16）－h：打印命令的帮助摘要；如果 dig 命令没有指定参数，则提供帮助摘要。

17）查询选项：dig 提供了许多选项，可指派带有包括或显式排除的查询特征。加号（+）被用来指明每个选项，no 关键字指明不使用指定的特征。就像在确认中输入的那样（没有 no 关键字），写上每个选项的描述。注意在可选的 no 关键字和选项名之间，在这个列表中给出一个空格（出于可读性考虑）。但是，当输入相应的命令时，要忽略空格，例如 + notcp 指明 + tcp 选项的相反指令（即 notcp）。

传输选项

1）＋bufsize = bytes：将 UDP 消息缓冲尺寸设定为 bytes 个字节；有效值范围从 0 到 65，535。

2）＋［no］fail：指令 dig 在接收到一个 SERVFAIL 结果时，不要尝试下一台候选服务器。这是 dig 的默认行为，这与解析器的行为相反。

3）＋［no］ignore：忽略从一条 UDP 查询得到的任何截短情况；正常情况下，这样的一个 UDP 截短场景将导致使用 TCP 重新发送该请求（当使用 + noignore 时出现这种情况），但使用 + ignore 设置的做法，就指令 dig 不要使用 TCP 重新发送该查询。

4）＋［no］tcp：使用 TCP 进行查询；默认情况下，对 AXFR 或 IXFR 查询使用 TCP，对所有其他查询使用 UDP。

5）＋time = time：将查询超时设置为 time 秒（默认 =5，最小 =1）。

6）＋tries = n：设置次数 n，在没有一条应答时，将发送一条 UDP 查询（默认 = 3，最小 =1）。

7）＋retry = n：设置次数 n，在没有一条应答时，在第一次查询后将重发一条 UDP 查询（默认 =2，最小 =1）。更清楚地说，就是，＋tries 指定总尝试次数，而 + retry指定在初始尝试之后的尝试次数。

8）＋［no］vc：使用 TCP 进行查询（vc = 虚电路）；默认情况下，对 AXFR 或 IXFR 查询使用 TCP，对所有其他查询使用 UDP。

解析器配置重写（覆盖）选项

1）＋domain = domainname：将域搜索列表排他地设置为指定的 domainname。

2）＋ndots = m：指定名字中要有的点数（即“.”）（m），如此才被认为是合格的；即，当在名字字段中输入较少的点数时，dig 将附加上在域中指定的域名或解析器配置中的搜索参数。

3）＋［no］search：支持基于解析器客户端指定的搜索列表或域指令（directive），进行域搜索列表处理。

4）＋［no］defname：废弃不用——等价于＋［no］search。

5）＋［no］showsearch：显示中间搜索结果。

DNS 首部设置选项

1）＋［no］aaonly：在查询的首部中设置权威答案（AA）比特，指明期望得到一个权威（非缓存的）答案。

2） + ［no］ aaflag：等价于 + ［no］ aaonly。

3） + ［no］ adflag：在查询的首部中设置真实的数据（AD）比特。这个选项用来提供“完备性”，虽然它并没有什么意义。正常情况下，AD 比特由一台服务器设置，指明查询响应数据已经通过 DNSSEC 核验得到验证。

4） + ［no］ cdflag：在查询的首部中设置检查去活（CD）比特，指令被查询的 DNS 服务器去活这条查询的 DNSSEC 签名核验功能。

5） + ［no］ dnssec：在 EDNS0 OPT 记录中设置 DNSSEC OK（DO）比特，指明期望进行 DNSSEC 处理。

6） + edns = version：将 EDNS 版本设置为 version。

7） + noedns：清除以 + edns 设置的 EDNS version。

8） + ［no］ recurse：在查询的首部中设置期望使用递归（RD）比特。除非指定 + nssearch 或 + trace 选项，否则 dig 查询默认地设置 RD 比特，请求递归查询。

输出选项

1） + ［no］ all：以默认格式显示结果；设置 + noall 会抑制所有结果。

2） + ［no］ cmd：显示 dig 输出的第一行，该行指明 dig 的版本和所施用的查询选项。默认地显示这一行。

3） + ［no］ comments：正常情况下，dig 显示以 DNS 消息格式组织好的结果，按照首部、问题节、答案节、权威节和附加节的顺序进行组织。应答的这些“节”和空行，被认为是输出中的注释，目的是提升可读性。设置 + nocomments 会抑制输出中的这些行。输出将仍然包含查询时间、服务器、时间戳和消息尺寸，虽然这些也可使用 + nostats 进行抑制。

4） + ［no］ identify：当与 + short 一起使用时，也显示 DNS 服务器（提供每个答案）的 IP 地址和端口号。

5） + ［no］ multiline：以多行格式显示复杂的资源记录（例如 SOA）；默认情况下，是在单行上显示一条记录。

6） + ［no］ nssearch：显示在 name 字段（或-n 参数）中所指定域权威服务器的 SOA 记录。

7） + ［no］ short：显示一个简洁的答案。例如，当发出查找一个给定名字的一条 A 查询时，仅显示所解析得到的 IP 地址（可能有多个地址）。

8） + ［no］ stats：显示查询时间、做出响应的名字服务器、时间戳和消息尺寸。

9） + ［no］ trace：针对所查询的名字，显示从根服务器到权威名字服务器的委派路径。dig 将向沿委派路径向下的每台服务器发出迭代式查询，显示沿路来自每台服务器的答案。在识别域树中有缺陷的（断开的）委派过程中，这是非常有用的。

DNS 消息选项

1） + ［no］ additional：显示响应的附加节（默认情况下，显示附加节内容）。

2） + ［no］ answer：显示响应的答案节（默认情况下，显示答案节内容）。

3） + ［no］ authority：显示响应的权威节（默认情况下，显示权威节内容）。

4） + ［no］besteffor：显示错误形式之消息的内容（默认情况下，不显示错误形式的响应）。

5） + ［no］cl：显示这条查询的 dig 结果的资源记录类。

6） + ［no］nsid：包括 EDNS NSID 请求选项，目的是从服务器处请求名字服务器的身份。

7） + ［no］qr：当将查询发送到 DNS 服务器时，显示该查询，默认情况下是按照首部和问题字段进行组织的。

8） + ［no］question：显示响应的问题节（默认情况下，显示问题节内容）。

9） + ［no］ttlid：显示这条查询的 dig 结果的资源记录 TTL。

DNSSEC 签名核验

1） + ［no］sigchase：寻找 DNSSEC 签名链；要求 dig 编译时使用-DDIG_ SIGCHASE 开关项

2） + trusted-key = filename：识别包含信任密钥的一个 filename（文件名），要与 + sigchase 一起使用；要求 dig 编译时使用-DDIG_ SIGCHASE 开关项

3） + ［no］topdown：当查找 DNSSEC 签名链与 + sigchase 一起使用时，实施自顶向下的核验；要求 dig 编译时使用-DDIG_ SIGCHASE 开关项

BIND 9 的 dig 工具允许在单一命令行上输入多条查询，方法是简单地串接后续查询的名字-类型-参数-选项字符串。例如，下面形象地说明了运行针对 IP 地址 10. 0. 3. 43 PTR 查找的查询，包括显示该查询，同时带有一条查找“ftp”的 CNAME 查询，还要将解析器域后缀设置为 ipamworldwide. com。

dig - x 10. 0. 3. 43 + qr ftp CNAME + domain = ipamworldwide. com

（3）DHCP 排错。可使用 DHCP 客户端能力，例如 Windows 的 ipconfig，或 Unix 或 Linux 的 ifconfig 命令，来实施测试 DHCP 事务。这些命令提供实施 DHCP 释放、刷新和设置用户类的能力。例如，在一个微软 Windows 上使用 ipconfig 命令行，支持使用如下参数显示 IP 配置：

1）/all：针对每个接口显示 IP 配置信息，包括以下内容。

① IPv4 地址和子网掩码。

② 其他 IP 地址，包括 IPv6 地址。

③ MAC 地址（可能有多个）。

④ 接口描述。

⑤ DNS 域后缀。

⑥ 默认网关。

⑦ DHCP 服务器，从之得到地址租赁，还要得到该租赁的日期/时间以及超期的日期/时间。

⑧ 用于解析器配置的 DNS 服务器。

⑨ 为进行 NetBIOS 查找，要查询的 WINS 服务器（如果配置了的话）。

2）如果忽略/all 参数，则仅显示 IP 地址、子网掩码、域后缀和默认网关。

3）/?：以命令摘要的形式，显示帮助（help）。

4）/displaydns：显示解析器缓存的内容。

5）/showclassid adapter：显示为指定接口适配器所配置的用户类。

ipconfig 也提供如下命令。

1）/release [adapter]：发出一条 DHCPRELEASE，释放所有或指定接口适配器的地址租赁。

2）/renew [adapter]：发出一条 DHCPRENEW，刷新所有地址租赁或指定接口适配器的地址租赁。

3）/registerdns：发出一条 DHCPRENEW，刷新所有地址租赁，并直接更新 DNS A 记录（可能有多个）（客户端发送到 DNS 服务器的，而不是 DHCP 服务器到 DNS 的）。

4）/flushdns：清除解析器缓存。

5）/setclassid adapter class：针对指定的接口适配器，设置用户类名。

14.5　记账管理

基本上而言，记账的意图是使每个人保持诚实。那些被指派的地址仍然在使用吗？任何没有被指派的地址实际上在被人使用吗？新子网在路由器上还没有被准备好可以提供吗？因此记账管理支持成功配置的验证，以及严格遵守 IP 编址规划。记账管理功能的各项技术，包括 IP 地址、路由器子网、交换机端口映射、DNS 资源记录和 DHCP 租赁文件等的发现分析。

被发现信息的分析是必要的，目的是将这个信息与 IP 清单“记录规划”进行比较。这项差异报告和比较过程是困难的工作，但却提供了一定程度的清单准确性保障。如果没有这项功能，无赖用户会免费地访问服务，或渗透到网络之中。另外，规划的网络变更还没有实现的话，则可导致下游处理延迟，以及服务准备提供间隔上的内部或外部服务水平协议的违规。

14.5.1　确保清单准确

到此为止我们所讲到的每项常见的 IP 管理任务，都依赖于准确的 IP 地址清单，来支持地址块、子网、IP 地址等的分配、删除和移动，以及 DHCP 和 DNS 服务器配置。对于这些地址管理任务而言，准确性是绝对必要的。但是对于通常的排错而言，准确的清单信息也同样是必不可少的。如果由于一次网络中断导致一个远程站点不可达，则为在该站点处的各设备识别 IP 地址、资源记录或其他 IPAM 有关数据，就是必要的。当最需要这样的信息以及不能直接从网络得到这样的信息时，仅有维护一个准确的 IP 清单，才能保证得到这样的信息。

在本节，我们将回顾这样的步骤，您可用来确保 IP 清单的准确性。这包括控制由谁对某些 IPAM 信息做出某些改变，发现实际的网络数据，将实际数据与清单保持一致，并最终回收地址空间。

（1）变更控制和管理员责任。在回顾这些 IP 管理任务中如我们所看到的，IP 清

单中的一次变更经常会影响其他网元，包括路由器以及 DHCP 和 DNS 服务器。如果不同人或团队管理着这些不同的单元，那么周期性地召开规划或变更控制会议，或依据需要来回顾和调度近期规划的寻址变更，就是一种好的思路。对这个过程要添加某种纪律性，要稍稍严格一些，并使那些可能被变更所影响的信息或设备处在控制之内。

有助于确保 IP 清单本身准确性的一种方式，是限制如下人员对清单的写访问，这些人员是 IP 寻址规划的权威，并相当了解该规划。使用单一口令保护的表格（仅有 IP 规划人员可修改），是保护 IP 清单不被疏忽的或错误的变更所修改的一种方法。但是，即使对于中等规模的组织机构而言，这种方法也是不可行的。如果组织机构依赖于单个人进行整个 IP 地址规划，那么这个人必须不停歇地工作，如果他或她离开该组织机构，那么恢复对清单的访问可能是非常困难的，除非提前培训一名接替人员。

并行支持多名管理人员是市场上多数 IPAM 系统的一项关键功能，且多数系统运行某种程度的范围控制，从而使某些管理员仅能在某些设备上或网络的某些部分实施某些功能。确保您所选中的系统支持管理员日志记录，这是预防如下情况，您需要查出在系统上“谁做了什么”操作。

和约束对 IP 清单的多名管理员受限范围访问一样重要的是委派职责，对 IP 地址指派、DNS 资源记录、子网地址等的任意变更，可在 IP 清单范围之外进行。例如，人工配置可能会输入错误，子网可能被配置在错误的路由器接口上，以及对 DNS 的客户端或 DHCP 更新，都可能导致 IP 清单偏离真实情况。IP 清单是 IP 地址规划的一个模型，IPAM 任务依赖于规划的准确性。因此，要从 IP 网络进行“规律性地读取数据”时，就是明智之举。周期性地轮询并将网络上的实际指派与清单进行比较，是确保清单准确的关键。

(2) 网络发现。有各种方法可用来采集网络真实数据，从 ping 到 DNS 查找到 SNMP 轮询。pinging 支持一个 IP 地址占有情况的检测，并提供哪些地址正在被用（与 IP 清单的相应部分进行比较）的一种基本方法。ping 是非常有用的，但要知道，一些路由器或防火墙将丢弃 ping 报文，或甚至一些设备可被配置成忽略 ping。将远程 ping 代理设置为依据命令实施局部的 pinging，这种做法可帮助绕开路由器/防火墙穿越问题。

可在 insecure. org/nmap 免费得到的 nmap，是以合适代价可得到的一个特别有用的工具。它组合了几种发现机制，用来从连接到 IP 网络的设备处采集各种信息，这几种机制包括 ping 扫荡（sweep）、DNS 查找和端口扫描。当扫荡一个子网时，nmap 可在一条命令中实施这些任务，向每个地址发出一条 ping，在 DNS 中查找一条相应的 PTR 记录，并尝试连接到各个 TCP 和 UDP 端口，以便识别设备的操作系统。从一个 IPAM 视角看，ping 结果有助于识别 IP 地址占有情况，DNS 查找帮助确认核实 DNS 服务器和 IP 清单之间主机名到 IP 地址的映射情况，端口扫描可提供占有每个 IP 地址之设备类型的其他信息。

SNMP 是发现 IP 清单有关信息的另一种方法。多数端设备（如笔记本计算机或

VoIP 电话）本地并不支持 SNMP，而多数基础设施单元（如路由器、交换机和服务器）却支持 SNMP。在路由器 MIB 内人们特别关注的是 Interfaces（接口）、IpAddresses（IP 地址）和 Arp 表。如果您的基础设施设备支持 MIB-II，那么在不同产品间这些表的解释应该是一致的。即使在来自同一厂商的不同产品间，也要了解微小差异。在这些表中的信息，支持接口和子网（每个接口所提供的）的信息收集，是由路由器报告的。一般来说，这提供了清单的有用核验方法，但也可在地址块和子网的分配、移动或删除过程中进行轮询。

轮询路由器的 ARP 缓存表，可提供在最近的子网通信上 MAC 地址到 IP 地址的一种确定映射关系。即使一台设备拒绝对一条 ping 做出响应，但也一定可以使用地址解析协议（ARP）形成一个层 2（例如以太网）帧，在该帧内封装预期包括的 IP 报文。正如这是被缓存的信息这个事实所蕴含的，它是临时性的，并必须频繁地加以轮询。

考虑到在一个/64 子网上有 2^{64} 个可能 IP 地址的庞大规模，所以要 pinging 一个 IPv6 子网是不现实的。轮询一个路由器的邻居发现表，例如 ipv6NetToMedia SNMP MIB，是实施 IPv6 主机发现的一种更加有效的方法。

（3）IP 清单一致性检查（reconciliation）。网络发现信息，提供了对实际的子网分配、IP 地址指派和相关联的资源记录等的真实检查。通过将发现的信息与 IP 清单数据库相比较，就可识别并调查差异。虽然这种比较会要求“仔细检查”（eyeballing）清单表格和发现输出之间的差异，但出于几个原因，证明这种付出是有益的。例如，可以识别数据库差异，可能是如下几方面导致的：

1）不正确的路由器信息提供。不正确的子网、掩码、路由器接口等。

2）不完全的路由器信息提供。规划的变更还没有实现。

3）设备可达性问题。如果一台设备应该处在一个给定 IP 地址上，但没有接收到响应。这可能源于一次设备中断、一次临时的中断（重启）、地址重新指派或网络不可达。

4）不正确的 IP 地址指派。人工配置的地址是不正确的，或设备从一个没有预料到地址池或地址处得到一个 DHCP 地址。这个问题特别适用于人工指派的地址，其中配置指派的 IP 地址、检测自动配置的设备和更新 DNS 时需要人工介入。

5）实际的 IP 地址指派。对于自动配置的设备，IP 地址是由设备选择的。同样在一些去中心化场景中，子网上一台设备的安装人员（installer）可选择一个 IP 地址。对于这些情形，可使用发现功能来更新 IP 清单。

6）信息不全的 IP 地址指派。无论是人工指派还是 DHCP 指派过程，指派过程的所有方面都是信息不全的。这个问题特别适用于人工指派的地址，其中配置指派的 IP 地址、检测自动配置的设备和更新 DNS 时都需要人工介入。

7）存在流氓设备。一个未知的或未被授权的设备已经得到一个 IP 地址。这提供了一种有效的访问后（post-access）控制机制，用来弥补和审计一种网络访问控制解决方案。

除了检测差异外，分析发现信息，可确认分配或指派任务以及删除任务的完成。

当移动地址块、子网和 IP 地址时，发现数据是必不可少的。因为移动要求分配新地址（可能有多个）、移动，之后是删除旧地址（可能有多个），所以在从 IP 清单中删除旧地址（可能有多个）之前，确认移动的完成就是至关重要的。在移动完成之前，这些地址不应该被删除，从而在它们实际被删除之前，使它们不会被未知地重新指派给其他设备或子网。

总之，对于确保 IP 清单的准确性而言，网络发现是至关重要的。对于监测信息准备提供或指派进度和时间帧、管理要求多个有关子任务的任务完成以及检测不正确的指派和潜在的流氓设备，网络发现也是有益的。

14.5.2 地址回收

上面所讨论网络发现和一致性检查的另一项益处是，设备可达性检测问题。如果一台设备已经准备就绪，并在一个给定 IP 地址上过去是做出响应的，但现在不再做出响应，这样的一个事件应该促发进一步的调查。如果没有计划移动或拆除该设备，或出现该子网上其他设备不存在网络问题，那么该设备可能遇到了一次中断，可能正在重新启动，可能被移动或断开了，或可能已经被重新编址。如果该服务器正在提供关键性的服务或应用，希望的情况是，您正在通过一个网络管理系统[⊖]监测它的状态，这种系统可证实停机推测，并触发正确的动作。如果在下一次尝试中发现了该 IP 地址，也许它只是简单地重新启动。如果在接下来的 n 次尝试中，它都没有做出响应，也许它物理上已经不再处在那儿了（或至少从电气角度已经不在那儿）。不幸的是，人们并不总是会通知 IP 规划团队，一台设备已经被拆除或移动到了其他地方，即使在最严格的组织机构中也是如此。一个快速的电话打到站点，检查该设备的状态，这种做法可能证明是成果显著的，但要确定设备的“主人”来验证状态的过程，经常是困难的和耗时的。

不过，评估该设备可能命运的关键点是，可能需要多次发现尝试，来确定一台设备是在那里还是已经不再了，是遇到了一次临时停机或断开，或被借走了但现在已经还回来了。跟踪连续发生的发现尝试可能是困难的。一个运行日志或表格可被用来对差异或“消失的”IP 地址（当它们被检测到［或不被检测到］时）作日志记录。随时间不同，检查这个日志，可帮助确定被记录为在使用的一个 IP 地址实际上并没有被使用的情况。

在检查这样的一个日志时，如果一个给定 IP 地址直到一个月前，还被成功发现，当时它是可达的，但在如此多的尝试（比如 30 次）之后，不可达了，则可确认该地址可用于未来的指派，或确认它是可回收的。回收的概念涉及识别 IP 地址，在 IP 清单中被表示为正在使用，但事实上没有被用，或在最近过去没有被用。分析多种发现结果的做法，提供了一种比较可靠的样本集合，在其上可做出一次回收决策，本质上是从清单中删除该设备，并释放地址，可指派给另一台设备。

除了对从 IP 清单中删除一台设备提供可靠确认外，回收也可类似地施用于子网。

⊖ 或如果该服务器是一台 DHCP 或 DNS 服务器，则可通过 IP 管理系统进行监测。

当删除一个子网时，一般来说，建议验证所有地址占用都被删除，并在该子网上不再使用 IP 地址[1]。分析一个给定子网上来自所有地址的发现结果，可提供该子网已被删除的确认保障。但像 IP 地址回收一样，多个样本集合可提供可回收处理的更可靠的确认。仅需要牢记的是，在一个子网上您将很罕见地看不到响应，至少在一个路由器接口上仍然配有地址，所以您需要检查后续的发现结果，忽略路由器、交换机，也许还有其他设备类型。

14.6 性能管理

性能管理涉及 IP 管理系统功能的监测，更重要的是，运行在网络中 DHCP 和 DNS 服务器的监测。跟踪基本的服务器统计信息，例如 CPU 利用率、内存、磁盘和网络接口输入/输出（I/O），是有用的。这样的监测，使跟踪硬件支持运行在服务器上 DHCP 和 DNS（和任何其他服务）的能力成为可能。在这方面的趋势分析，在支持未来硬件采购的先验规划方面也是有益的，目的是支持在多台服务器间的负载均衡分布。

14.6.1 服务监测

对 DNS 服务的监测，有助于确保充分的 DNS 能力来满足名字解析的需求，并有助于识别任何意外条件。BIND 支持将各种事件类型灵活地记录日志到一个可配置的输出目的地或通道，包括 syslog、file、null 或 stderr（操作系统的标准错误输出目的地）。微软支持 DNS 服务器事件查看器，带有可设置的严重性等级报告以及接收到的总查询数/秒和发送的响应数/秒等统计计数。DHCP 服务器类似地提供监测总体服务健康程度和统计的日志记录，典型地记录到一个日志文件、syslog 或一个事件日志。

这些措施支持从服务器视角的性能数据收集。但是，这些措施并不像 DHCP 客户端和 DNS 解析器所经历的那样传递服务性能。测量客户端性能，要求远程发出一条 DNS 查询或 DHCPDISCOVER（或 SOLICIT）报文，并测量接收到一条正确响应的响应时间[2]。这种远程发出报文的做法，可能源自于部署于各种位置的服务探针（probe），可产生这些“合成的事务”，由探针测量并存储响应时间结果。对来自不同探针的历史数据的分析，可提供对 DNS/DHCP 服务和网络性能的敏锐理解。

14.6.2 地址容量管理

总体 IP 地址容量监测，是 IPAM 范围内另一项关键的性能管理功能。对人工编址设备和通过 DHCP 得到地址的那些设备的地址利用率，进行跟踪，可支持地址空间的先验管理。地址管理最初是依据估计预测的，人们希望它是准确的。但是，即使在预测是完美情况下，由于雇员调动、大型事件、订户增长以及未规划的地址需求等导致

---

[1] 忽略路由器 IP 地址占用，原因是典型情况下路由器在子网上进行自我识别。

[2] 如在故障管理节所提到的，一条响应的缺位，可能表明一次服务停机，如果持续出现这种情况的话，就应该进行调查。

的 IP 网络动态性，可消耗一个子网及其地址池的整个容量。对地址池和子网利用率水平的周期性监测，以及历史跟踪和趋势分析，可提供容量耗竭的预先告警，触发补偿性的分配，以便在容量用光之前扩展容量。

许多 DHCP 服务器产品支持对地址租赁水平进行监测，采用的是命令行、脚本或 SNMP 方法。微软 DHCP 也提供通用的 90% 告警阈值，如果一个地址池达到 90% 容量，就提供通知。其他服务器和 IP 管理系统提供类似的或附加的告警阈值定义和应用。由一个 IP 管理或网络监测系统进行通知，通常比起尝试访问网络的暴怒顾客或端用户进行通知而言，要好得多。

14.6.3 审计和报告

一般而言，多数管理系统提供某种程度的“谁在做什么”的审计和不同等级的报告。这些功能，就和被分类在记账管理之下一样容易，使管理员可跟踪并排查活动，并以报告形式传递状态信息。当对一个网络问题排错，或调查可能的违规活动时，对 IP 地址使用情况的审计（即谁在某个时间点拥有一个给定的 IP 地址），是有价值的信息。类似地，如果您尝试跟踪一台给定设备的 IP 地址占用历史，那么按照硬件地址进行报告也是有益的。

除非对于最小型的网络，否则在没有一个 IP 管理系统的情况下，实施这样的审计可能是困难的。重复地对 DHCP 租赁数据进行导出（dump），以便跟踪随时间消失而发生的动态编址客户端的做法，是必要的。对一个给定 IP 地址搜索的能力，要求访问单一（或者采用故障切换或事实上分割地址范围的话，就是两台）DHCP 服务器的地址租赁历史，而依据硬件地址进行搜索，则在所有 DHCP 服务器间进行搜索就成为必要的，这里假定了设备是能够移动的。

对 IP 地址规划有用的常见报告包括如下内容，虽然您的系统可提供不同的或附加的报告。

（1）地址利用率报告。按照地址池、子网、地址块进行报告，是按照层次结构逐层报告的。

（2）地址指派报告。依据子网或地址块方式，对指派地址的摘要，为当前快照和/或历史情况。

（3）地址差异报告。重点突出 IP 清单和所发现 IP 地址信息之间的差异。

（4）DHCP 性能报告。依据类型的 DHCP 消息的摘要和细节，和/或客户端和服务器主要的指标摘要。

（5）DNS 性能报告。依据类型、查询器、问题等的摘要和细节，以及服务器主要指标摘要。

（6）审计报告。依据子网、IP 地址、硬件地址、资源记录、服务器等的管理员活动。

14.7 安全管理

数个 IP 管理厂商已经尝试加入"NAC"事业，使 IP 规划人员将 IP 地址指派仅限制到有效的设备或用户，这是由 DHCP MAC 地址过滤或用户登录确定的。如果一台设备或用户多次尝试失败，那么这种解决方案需要提供意外报告，理想情况下提供告警。一次失败的访问尝试可能归咎于错误输入，但数次失败可能表明，一名攻击者正在破解系统以便访问网络。当然，如果该攻击者知道子网地址，他/她可手工地配置一个 IP 地址来访问网络，旁路掉 DHCP。在第 8 章中已经和其他方法一起，详细讨论了这个过程。

第 8 章、第 12 章和第 13 章分别处理 DHCP 和 DNS 服务器信息的安全保障以及与这两种服务器的事务过程安全保障。保障 IP 清单以及 DHCP 和 DNS 配置数据的安全也是重要的，原因是这种信息是至关重要的，并应该保护它使之不受蓄意破坏。保护 IP 数据，要求管理员访问控制，至少要支持对信息的口令保护访问。如果您的组织机构有两个或三个以上的管理员，那么通过使用一个 IP 管理系统，可确保使用一种比较复杂的管理员安全方法。多数系统采取的措施远不止最低限度的口令项，它们支持依据系统功能、依据网络部分或某些单元、依据访问类型（比如），将管理员访问限制为超级用户、只读访问以及其他权限。访问控制的复杂性将受到管理员数、管理员的角色和职责以及组织机构的安全策略的驱动和影响。

14.8 灾难恢复/商务持续性

商务持续性实践，在面临一次重大的停机时，寻求维护企业的运作。一次大型停机或"灾难"意味着远多于数台服务器或网络设备的大规模停机运转。必须提前记录归档自动化的和人工过程，是为了维护网络和应用的运行，或至少是关键服务和应用的运行，需要重新配置或重新部署资源。在第 7 章和第 11 章，我们讨论了部署和配置冗余 DNS 和 DHCP 服务的各种方法。一旦部署和配置，在出现个体服务停机时，冗余功能应该提供网络服务持续性。在这些技术之上，DHCP 和 DNS 仪器设备的部署应该提供一个附加的可用性层。典型情况下，虽然这提供站点内硬件冗余，但这种模型通常情况下并不被看做灾难恢复解决方案，原因是这种模型要求处于相同位置。不过，仪器设备冗余提供可靠的冗余选择，特别对于关键的服务器（例如主 DNS 服务器）尤其如此。

IPAM 运行的商务可持续性，将可能要求部署多个 IPAM 数据库。多个活跃的数据库或主/备份配置的部署将取决于您所选中的厂商。各厂商实现各种方法，以便实现冗余（如全数据库拷贝和传递，多个主数据库（要求某种程度的网络分隔））到数据库复制技术的部署，其中使用存储区域网络或 SQL 或 LDAP 复制能力。实施一次灾难恢复所要求的操作任务，将可能随厂商而不同。

对厂商灾难恢复能力的评估，将取决于您的商务目标、预算和策略（polices），

但应该考虑三个关键问题。

（1）IPAM 数据库涉及名字解析或地址指派吗？例如，一些系统将来自 DHCP 的动态更新，路由到 DNS 之前，首先路由到 IPAM 数据库，进行唯一性检查。如果 IPAM 数据库处在这样的一个“关键路径”中，那么冗余和高可用性就是极端重要的。

（2）您的管理员们对 IPAM 数据做出改变的频率有多频繁？改变的速率越高，则在主数据库和备份数据库（可能是多个）之间的两次数据同步间，可能丢失的数据就越多。在改变不太频繁的情况下，每天进行数据库备份就是一种可接受的方案，而对于高改变频率的环境，可能要求小于一天的或事务级复制过程。

（3）实施备份系统触发的过程是什么？一些厂商提供一个相对简单的回复过程，而其他厂商则要求更多的人工介入。考虑到一次灾难恢复故障切换的广泛影响，人们也许期望某种较小程度的人工介入。间歇性的问题可能导致假阳性（false positives）结果和潜在的突然的故障切换，所以人工地启动故障切换，可能会消除由解决方案所导致的灾难。人们希望，如果不会永不发生，那么灾难也是不频繁发生的，但当需要故障切换时，也应该在策略所定义的时间约束内由员工手工地执行故障切换过程。

这些基本问题是相互依赖的。如果问题 1 和问题 2 的答案分别是“是”和“频繁的”，那么问题 3 的答案也许是“单步骤的”或至少是“非常高效的”。

14.9 ITIL 过程映射

IT 基础设施库[⊖]是期望管理、监测，并持续改善企业组织所提供 IT 服务质量的一个 IT 组织机构所用的最佳实践文档集合。ITIL 是由英国商务部（U. K Office of Government and Commerce）开发的，其面向 IT 服务的方法已经由许多组织机构进行部署实施。ITIL 实现的最常见驱动因素包括以下内容。

1）IT 服务交付到组织机构的成本降低。

2）通过潜在的影响服务变更的有纪律性的规划和评估，进行风险管理。

3）IT 服务水平一致性和改善。

4）利用有文档记录的过程和持续改进，所得到的效率。

在这些过程域中的许多功能是等同的或类似于前面章节中讨论的那些功能，所以我们将在其 ITIL 过程域的对应映射中简单地总结描述这些功能。

14.9.1 ITIL 过程域

ITIL 过程分为两个域：服务交付和服务支持。服务交付涉及 IT 服务的规划、开发和部署，而服务支持包括管理服务水平和支持中的各项服务操作。服务台（service desk）是第三方过程域，它将服务交付和服务支持进行集成，向组织机构的其他部门

⊖ 在 ITIL 网站 http：//www. itil - officialsite. com/home/home. asp[168] 可找到细节。本节中的功能映射依据的是参考文献［174］中最初讨论的那些内容。

提供 IT 的一个统一的接口。ITIL 版本 3 构建于这些过程集之上，带有一些添加内容和功能分割。

(1) 服务交付。服务水平管理是一个服务交付过程域，它包括 IT 组织机构所提供各种服务的服务水平规范。这与服务水平协议（合同）相近。服务水平管理也包括依据这些规范进行服务交付的度量，以便监测与 IT 所提供服务水平的吻合性，并测量服务水平。

从一个 IPAM 角度看，服务水平管理可能涉及服务水平的定义和度量，这些服务是提供给请求 IPAM 有关服务的那些实体，不管它是请求一个 IP 地址的端用户或需要开设一个新的零售办事处的商务实体。将这些请求中的端用户或商务实体看作顾客，这个过程寻求测定服务交付是否满足确定的服务水平，例如这些请求完成的及时性。使 IPAM 有关的服务交付自动化，不管是仅有 IPAM 发挥影响或涉及 IPAM 作为一个较大型 IT 服务的组成部分（例如 VoIP 部署），都有助于及时的和准确的服务交付。一个范例度量指标是时间帧，据此可指派一个子网或一个 IP 地址。

财务管理自然地包括记账，这类似于 FCAPS 模型中的记账管理，虽然财务管理域处理的是实际的金钱（dollars and cents as well）。这个过程域也处理在一个 IT 资金或成本分配模型中某些部分的任何回冲（chargebacks）或成本分配。

一些公司确实实施 IP 地址使用情况的成本回冲（chargeback）。在这样的场景下，财务管理过程将需要记录跟踪 IP 地址使用情况以及相应的用户和可收费的实体（例如部门）。取决于计费或回冲（chargeback）周期，这种 IP 地址使用信息将需要针对当前周期进行存储，可能存储更长时间，目的是支持归档或争议解决。在证明成本分配合理性方面，您的 IPAM 系统中的审计和历史数据，也可能是一项巨大帮助。

容量管理简单地涉及确保正确类型的充足 IT 资源，可用于商务实体来实施它的工作。考虑将这种概念应用到 IPAM，当然 IP 地址容量管理会出现在脑海中，但人们也应该考虑 DHCP 和 DNS 服务器负载容量。在前一种情形中，容量管理要求监测地址和地址池，以便为雇员们得到一个地址并访问网络提供足够的 IP 地址。对趋势进行监测也是有帮助的，同时对于严格分配的网络而言，建议出现较低地址池时提供告警。当然，考虑到 IPv6 地址空间容量，在可预测的未来，对于 IPv6 空间这不太可能会成为一个问题。

就服务器容量管理而言，随时间推移监测每台服务器的网络、内存和 CPU 利用率情况，可提供对服务器性能的深邃理解。可能实际上要求实施这种性能任务，原因是就事务完成百分比（地址租赁或解析）以及响应时间而言，它与服务水平管理是有联系的。不仅如此，服务器上的过量负载对 DNS 和 DHCP 服务的可用性具有有害影响，所以服务器监测，也许甚至是类似探针的事务型监测，可提供服务水平和容量的实际测量。

可用性管理是这样一个服务交付过程域，它将焦点放在确保 IT 服务对端用户是可用的。高可用性，这包括 DHCP 和 DNS 等应用的一个共同目标，要求冗余配置的

部署和利用这些配置的能力，在面临一个组件停机时提供持续的服务。

如在第 7 章和第 11 章所讨论的，冗余仪器设备的部署，可提供局部化的集群，与 DHCP 故障切换或分割范围法的实施以及多服务器 DNS 部署一起，提供一个附加的冗余层。通过 LDAP 或复制的关系数据库方法，冗余 IPAM 数据库部署也可确保管理 IP 空间的 IPAM 应用的可用性。对每个这种冗余组件可用性的监测，支持停机的先验性检测，以便有助于快速的停机解决问题（修复的平均时间），同时冗余组件会承担负载。

持续性管理与可用性管理有关，其中它处理以持续方式提供可用的服务。例如，出现一次灾难时，这个过程域将要求提前准备好的一个灾难恢复计划。如在前面的商务持续性一节所讨论的，基于组织机构的关键性需求和范围需求，特别是对 DHCP/DNS 服务器和 IPAM 系统而言，存在各种战略措施（strategies）。

（2）服务台。作为用户团体的接口，服务台将到 IT 组织机构的意外报告、变更请求以及甚至新的服务请求等的输入进行过滤。它用于过滤并将用户请求或问题定向到其他 ITIL 域中的任何一个域，为端用户提供一项帮助台功能。

组织机构的策略和文化，将驱动服务台实施传统的“等级 1”支持（仅对后续步骤的问题进行日志记录），或实施较高支持等级（直到前台处理，实施一定等级的诊断）。在等级 1 支持的情形中，相比于一种票据（ticketing）系统，需求几乎并不多什么，票据系统具有这样的能力，即将票据指派到负责其他过程域的那些实体，这取决于呼叫者出现什么样的问题。一个服务台配备雇员，实施一些问题诊断，这将要求对状态监测工具的访问使用，就问题方面，尝试“看到呼叫者所看到的”情况。

对于 IP 地址或名字解析有关的呼叫，为服务台员工提供到 IP 清单信息的访问能力，可能证明是有益的。例如，如果位于费城总部办事处的一个人，不能得到一个 IP 地址，服务台需要知道总部的地址规划，以便将问题和问题解决过程集中到那个特定的子网、关联的路由器或 DHCP/DNS 服务器。

服务台不仅是问题报告的接口，而且是变更请求（例如 IP 子网或地址指派）的接口。为服务台员工提供到 IPAM 系统请求这种变更的基本访问能力，或更好的做法是，使端用户自己将这种服务请求注册到一个自动化的 IT 门户，这可通过快速解决问题（rapid fulfillment）增加端用户对 IT 服务的满意度。

（3）服务支持。意外管理是这样一个服务支持过程域，它涉及跟踪并解决意外情况。在 ITIL 版本 2 中，它也处理变更请求，而在版本 3 中，这些被分割成独立的过程域。如上面所描述的，服务台直接从端用户团体接收意外和变更请求。通过网络监测，IT 也可先验地检测和排除网络问题。不管一个给定意外情况所采取的检测方法为何，对 IP 清单数据的访问能力与排错和意外解决是不可分的。另外，以阈值、告警、记录日志信息和审计等来监测服务器状态，可提供意外检测和管理的一种良好开端做法（head start）。

服务请求可如下处理，即通过服务台或 IT 万维网门户发起服务请求票据，这种做法见服务台一节所描述的内容。

问题关联要求跟踪一个已知问题数据库中的已知问题和解决方案。例如，如果某个人呼叫服务台，声称出现一次意外情况，可直接进入问题管理，识别这个意外在过去是否已经有人报告并得到解决。如果情况是这样的话，那么可遵循确定的解决路径，快速地排错并解决问题。

虽然IPAM系统传统上并不将问题历史与解决批注（annotations）存储在一起，但一些系统通过日志记录历史和清单变更审计，提供问题信息的一个数据库。通过API的网络管理系统集成，可提供问题历史的一个整体（holistic）视图，方法是采用API向一个问题票据系统提供IPAM数据（例如）。IPAM是整体网络或IT服务管理方法的一个关键组成部分，但它是不完备的；没有系统是完备的。采用那种集成法，是拥有问题管理范围一个整体视图的一把钥匙而已。

在ITIL内的配置管理，就识别、记录和控制影响IT服务的配置参数而言，类似于FCAPS配置管理功能。如我们在本章所大量讨论的，配置管理功能是IPAM过程的一个重点关注的焦点域。这包括配置新的地址池（从一个DHCP角度而言）、在DNS中配置区域和资源记录、在路由器上为子网配置IP地址，等等，所有这些都落在配置管理的疆域内。IPAM数据库可被看做IT用于网络配置请求的CMDB联合体（confederation）的一个配置管理数据库（CMDB）组件。

对于配备一名以上的IPAM管理员的组织机构来说，需要考虑实施管理员控制，以便确保对DHCP和DNS配置的变更是在合适范围和许可权下进行的。例如，您可能想让管理员们能够做出改变，但并不实际上在DHCP和DNS服务器上部署这些改变——将部署改变的功能限于一个较高等级的管理员。在后台（back end），审计信息是责任跟踪和报告的关键。

拥有准确的IP配置信息是必要的，这可为可规划的未来配置变更，提供一个坚实的基础。一个推论性的需求，导致将那个清单与网络实际数据进行核验的必要性。在一个表格上跟踪IP清单是不错的，但这要求不断地更新。从网络收集信息，并之后将这个信息与规划内容进行比较，这种能力是非常重要的。审计与清单信息收集是同样重要的。为服务台配备这种信息，可为立即解决呼叫问题提供坚实的第一道防线，或至少将呼叫快速地移动通过处理过程。

变更管理提供在IT基础设施中实施变更的控制。这涉及就所提变更的范围和实施时间方面，确保所有被影响的各方是一致的。就IPAM而言，常见的情况是，变更管理的范围会影响IPAM组件，例如增加一个地址池，在网络中部署一台新的DHCP/DNS服务器，或将一台服务器升级到一个新的软件版本。基本上而言，影响基础设施任何部分的任何事情，无论是物理的或软件的或甚至是低层仪器设备的操作系统，都落在变更管理过程之下，这个过程寻求确保所有合适的批准都就位，且有可用的相应退出（backout）计划。

发行管理是这样一个服务支持过程域，它对硬件和软件版本的部署发行提供控制，不仅针对操作系统，而且对应用和仪器设备都是如此。这个过程域负责在IT网络上，那些版本是可得到的和可访问的，并确保存在发行和版本的一个授权集，可被用来进行合适的部署。

通过一个中心地点，对 DHCP 和 DNS 服务器的发行规划和升级管理，会是一种巨大的时间节省方法（timesaver）。这种替代方法要求操作系统、补丁和应用软件的现场（on-site）升级是成本高昂的和消耗时间较大的。IPAM 的发行管理也落在这个分类范围内。

14.10　结论

FCAPS 和 ITIL 是类似的，是指它们都提倡带有严格约束执行的文档化的过程。IPAM 对一个组织机构的重要性，应该驱动将 FCAPS 或 ITIL 原理应用到组织机构内的 IPAM 实践。本章给出了一个 FCAPS 框架上下文中主要 IPAM 过程步骤的详细视角，还给出了到 ITIL 的一种建议映射关系。

第 15 章 IPv6 部署和 IPv4 共存

15.1 引言

IPv6[1]最初是在 20 世纪 90 年代中期规范定义的，目的是为了解决弥补快速耗竭的 IPv6 地址空间的当时迫切需求。当时在定义 IPv6（或 IPng（IP 下一代），在最初命名时是这样的）的工作上是在巨大热情中开始的，当时因特网刚刚开始在普通公众中流行。越来越多的企业扩展它们的内部网，以便支持到全球因特网的连接。因为每台可达的主机都要求一个唯一的公开 IPv4 地址，所以对地址的需求猛涨。

这些事件激发了 IPv6 的开发，它们也刺激了其他技术的开发，这些技术延长了 IPv4 地址空间的可预期寿命。如我们在第 1 章所讨论的，无类域间路由（CIDR）使区域因特网注册机构和 ISP 能够更加高效地分配地址空间，这是相比以前仅以有类（classful）分配方法而言的。

在第 1 章中讨论的另一项 IPv4 分配策略是分配私有地址空间，这种方法极大地降低了组织机构从 RIR 或 ISP 要求的地址空间量。在 RFC 1918 中定义的私有网络分配，使每个组织机构能够为它们的内部网络使用相同的地址空间。与因特网主机的通信或组织机构间的通信，仍然要求公开地址，但网络地址转换（NAT）防火墙的使用，为内部主机访问因特网提供了私有地址到公开地址的转换。

人们可能会争辩说，DHCP 本身也可支持地址空间的更好利用，它具有依据需求而在多个用户间共享地址的能力。虽然在组织机构内普遍配置有私有地址空间，对公开地址空间几乎没有什么影响，但 DHCP 也为宽带和无线服务提供商所用，目的是支持其相应用户基数的因特网访问。

这些日益增长的用户基数仍然驱动着增长的 IPv4 地址消耗，这削减了可用的容量。在世界各地，IPv4 地址空间的当前分配是不够的。例如，亚洲被分配了大约 20% 的 IPv4 地址分配，但却支持了世界人口的 50%。在 IPv6 部署方面，亚洲是领先者之一，之后紧跟的是欧洲。虽然北美拥有相对充足的 IPv4 地址空间，但许多组织机构正在评估迁移到 IPv6 的工作，特别政府和服务提供商组织热心这种活动。

当我们讨论 IPv6“迁移”时，我们指的是仅支持 IPv4 网络的一个初始状态，随时间添加了或重载了 IPv6 节点和网络，得到一个仅支持 IPv6 的网络，或在多数情形中更可能的是，一个普遍存在的 IPv6 网络带有持续的 IPv4 支持。

---

[1] 本章中的材料依据的是参考文献［11］的第 8 章和参考文献［147］，但增加了更多细节。

15.1.1 为什么要实现 IPv6

在 2007 年 5 月 22 日，美国因特网地址注册机构（ARIN——RIR 之一）委员会发出一份公开声明“建议迁移到 IPv6 编址资源的因特网团体，要求从 ARIN 得到连续 IP 号码资源未来可用性的任何应用，是一个必要过程”。这本质上声明，IPv4 编址资源（包括 IPv4 地址）正在变得日益稀缺，随着时间推移而要求附加地址的实体（LIR、ISP），需要计划 IPv6。所有 RIR 也发表过类似声明。一些估计预测在大约 2012 年左右 RIR 层次会发生 IPv4 空间耗尽[⊖]。事实上，APNIC 的 Geoff Huston 博士在 www. potaroo. net/tools/ipv4 发布了有启发作用的分析，是每天都更新的，提供了这些悲观预测的证据。

在 2012 年 IPv4 空间的这种最终分配到来时，或无论何时发生这种情况时，ISP 们将仍然有地址空间可分配；由一个 RIR 所分配的最后一个地址块将耗尽 RIR 空间，但这个已分配的 ISP 地址块仍然可用于 ISP 分配。当 ISP 客户基数完全消耗掉他们的可分配空间时，ISP 空间将耗竭，在这之后将需要一年左右的时间。没有提交请求 IP 地址而且在可预见的将来没有计划提交请求的企业，可能得出结论，IPv6 将不会影响到他们。但在这个时间点，仅有 IPv6 空间可用的情况下，新的 ISP 或寻求扩展地址空间的组织机构，将为端用户生成仅有 IPv6 连通的情形。随着这种仅支持 IPv6 用户群的增长，这将驱使仅支持 IPv4 的组织机构在外部（面向因特网）万维网、电子邮件和其他公开服务器上实现 IPv6。

另外，像许多 IP 网络变化（例如由在组织机构内采用无线和 PDA 的那些变化）一样，IPv6 最终将会由通过下一代蜂窝电话、PDA 连接到网络的雇员或甚至通过网络或家庭连接的 Windows Vista（或 Windows 7）得到推广使用。在这种情形下，管理 IPv6 地址空间的需求会强加到 IT 管理员身上。无论是由外部仅支持 IPv6 的用户还是内部用户所驱动的，现在就提前针对 IPv6 做出计划，就是比较深谋远虑的做法，无论您计划完全地迁移您的网络，或计划处理尝试连接的个体 IPv6 设备，都是如此。许多服务提供商已经稳步踏上了在其 IP 网络上迈向 IPv6 部署的道路。

15.1.2 IPv4-IPv6 共存技术

存在许多种技术，可便利设备迁移到 IPv6。因为“迁移”将可能是一个非常漫长的过程，所以我们使用术语“共存”来体现这种情况。我们将依据如下基本分类，讨论这些方法。

（1）双栈。在网络设备上同时支持 IPv4 和 IPv6。

（2）隧道。在一个 IPv4 网络上进行传输时，将一条 IPv6 报文封装在一条 IPv4 报文；或相反的情况亦然。

（3）转换。IP 首部、地址和/或端口转换，例如有网关或 NAT 设备所实施的那

⊖ 虽然人们可从这个分析中推测 IPv4 地址空间耗尽的日期，但 Huston 博士分析的意图更多地将焦点放在识别何时 RIR 策略才能到位，来处理管理 IPv4 地址资源的新 RIR 角色。

些功能。

所选中的策略要求如下方面的有效协同。

（1）IPv4 和 IPv6 网络及子网分配，无论是现有的还是计划中的。

（2）验证网络基础设施和应用是与 IPv6 兼容的。

（3）对于期望的隧道法或转换法，DNS 资源记录配置对应于合适的名字到地址（可能是多个）解析。

（4）在合适的情况下，对所选中隧道模式的兼容客户端/主机和路由器支持。

（5）在合适的情况下，部署转换网关（可能是多个）。

在本章后面，针对服务提供商和企业，给出范例 IPv6 实现场景，但首先让我们讨论关键的共存技术。

15.2 双栈方法

双栈方法是这样组成的，在要求访问两种网络层技术的设备上，实现 IPv4 和 IPv6 协议，这些设备包括路由器、其他基础设施设备、应用服务器和端用户设备。这样的设备将配置有 IPv4 地址和 IPv6 地址，且它们可通过为相应协议定义的方法（由管理员激活）来得到这些地址。例如，一个 IPv4 地址可能是通过 DHCPv4 得到的，而 IPv6 地址可能是自动配置的。

就协议栈的范围而言，实现可随双栈方法而发生变化，协议栈是共享的或对每个 IP 版本是不同的。理想情况下，仅有网络层是双份的（dualized），使用共同的应用、传输和数据链路层。这是微软 Vista 和微软 7 中实现的方法，这与 XP 实现是相反的，后者利用双份的传输层和网络层，这在一些情形中，要求进行每个栈的冗余配置。其他的方法可能跨越整个栈，直到物理层，都要求对 IPv6 和 IPv4 使用独立的网络接口。这种方法，虽然与一种分层模型的益处相对，可能是有意而为且甚至是人们所期望的，特别在网络服务器带有多种应用或服务的情况下尤其如此，其中一些应用或服务可能是有意地仅支持一个版本或另一个版本。

15.2.1 部署

部署共享共同的物理网络接口的双栈设备，意味着在同一物理链路之上同时运行 IPv4 和 IPv6。毕竟，幸亏由于协议分层，以太网和其他层 2 技术才可支持 IPv4 或 IPv6 净荷。双栈设备要求支持这种链路的路由器也是双栈的。在转换过程中，人们预期这种重叠方法是非常普遍的，并如图 15-1 所示。这个图可扩展到一个物理 LAN 之外，扩展到一个多跳网络，其中路由器支持 IPv4 和 IPv6，并在纯粹 IPv4 主机间路由，在支持 IPv6 的主机间路由 IPv6 报文。

虽然通常情况下，人们预料路由器将是被升级支持两种协议的首批 IP 网元，但 RFC 4554[193] 是一个信息型的 RFC，其中描述使用 VLAN 的一种创新方法，在不要求立即升级路由器的条件下，支持一种重叠配置。这种方法依赖于 VLAN 标签，使层 2 交换机能够广播或捆绑传输（trunk）包含 IPv6 净荷的以太网帧，传输到一台或多台

支持 IPv6 的路由器。通过将一台路由器升级来支持 IPv6，例如到一个 IPv6 网络的网关，那么交换机接口被连接到的路由器接口，可配置为“IPv6 VLAN”。之后其他 IPv6 或双栈设备可配置为 IPv6 VLAN 的成员，且多个这样的 VLAN 可进行类似配置。这种部署的一个例子如图 15-2 所示。

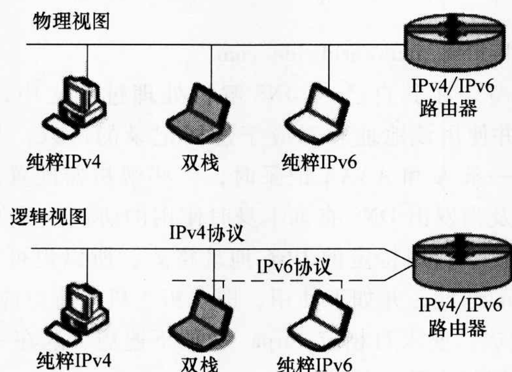


图 15-1　双栈网络视图[11]

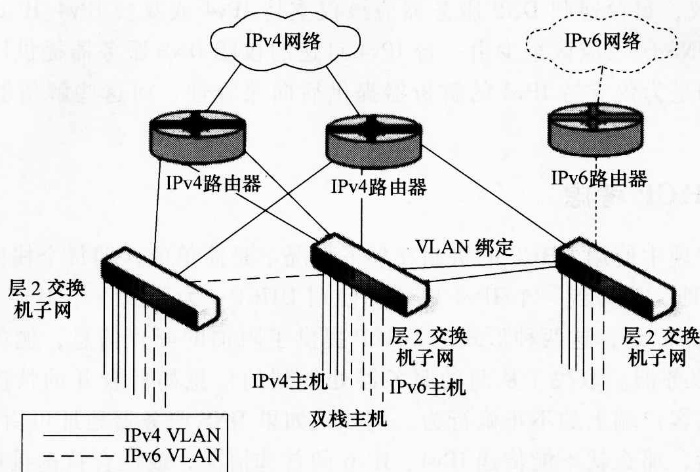


图 15-2　使用 VLAN 的双栈部署[147]

15.2.2　DNS 考虑

如我们将看到的，在每种迁移技术的正确运行中，DNS 扮演了一个至关重要的角色；毕竟，它提供了端用户命名（例如在应用层的网站地址）和目的地 IP 地址（在网络层的 IPv4 或 IPv6 地址）之间的重要联系。尝试访问一台双栈设备的端用户将查询 DNS，管理员可配置该 DNS，配置对应于节点 IPv4 地址的一条 A 资源记录和对应于其 IPv6 地址的一条 AAAA 资源记录。资源记录的属主字段可能有对应于设备的一个常用主机域名，见如下例子。

dual-stack-host. ipamworldwide. com. 86400 IN A 10. 200. 0. 16

dual-stack-host. ipamworldwide. com. 86400 IN AAAA 2001：DB8：2200：：A

IP 地址到主机域名的解析，也可配置在合适 . arpa 域内的 DNS 之中。

16. 0. 200. 10. in-addr. arpa. 86400 IN PTR dual-stack-host. ipamworldwide. com.

A. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 2. 8. B. D. 0. 1. 0. 0. 2. ip6. arpa. 86400

IN PTR dual-stack-host. ipamworldwide. com.

一个双栈节点本身，在其自己的 DNS 解析处理过程之中，必须能够支持 A 和 AAAA 记录的接收，并使用该地址和对应于返回记录的协议，与预期目的地进行通信。当从查询中返回一条 A 和 AAAA 记录时，一些解析器配置可能支持首选网络协议的定义，而并不涉及当发出 DNS 查询本身时使用的协议。另外，如我们将看到的，一些自动化的打隧道技术利用特定的 IPv6 地址格式，所以也可能返回对应于一种或多种隧道化地址格式的地址，并如下使用，即解析主机支持对应的隧道化技术。解析相应的 PTR 记录的做法，要求对相应 . arpa. 树向下遍历，这在一些情形中可能有误，即带有欺骗性质（我们将在下面讨论）。

就 DNS 查询和答案的传输中所用的 IP 版本而言，RFC 3901（因特网当前最佳实践 91）[148]建议，每台递归 DNS 服务器应该仅支持 IPv4 或双栈 IPv4/IPv6。该 RFC 也建议，每个 DNS 区域应该至少由一台 IPv4 可达的权威 DNS 服务器提供服务。提出这些建议，目的是为仅支持 IPv4 的解析器提供后向兼容性，而这些解析器将存在相当长的时间。

15.2.3 DHCP 考虑

在双栈实现中使用 DHCP 的机制在如下情况下是简单的，即每个栈使用其对应的 DHCP 版本。即，为得到一个 IPv4 地址，使用 DHCP；为得到一个 IPv6 地址或前缀，使用 DHCPv6。但是，这两种形式的 DHCP 提供了附加的配置信息，例如要使用哪台 DNS 或 NTP 服务器。取决于从两种服务器处得到的信息如何合并的情况，所得到的信息可能导致客户端上的不正确行为。例如，如果 DNS 服务器地址可由这两种 DHCP 事务提供的话，那么就不能传递 IPv4、IPv6 的首选情况，或混合首选排序信息。这仍然是人们关心的一个待决领域，在 RFC 447[149]中进行了记录，但当前标准是为 IPv4 使用一台 DHCP 服务器，为 IPv6 使用一台 DHCPv6 服务器，它们可能实现在一台共同的物理服务器上。

15.3 打隧道方法

已经开发各种打隧道的技术，用来支持 IPv4 之上的 IPv6 和 IPv4 隧道之上的 IPv6。一般而言，这些技术被分类为配置的或自动化的。配置的隧道是预先确定的，而自动化的隧道是在线产生和拆除的。在回顾一下打隧道方法基础知识之后，我们将讨论这两种隧道类型。

一般来说，在一个 IPv4 网络上以隧道方式传输 IPv6 报文的做法，涉及在一条

IPv6 报文前面打上一个 IPv4 首部，如图 15-3 所示。这使打过隧道的报文能够在一个 IPv4 路由基础设施上被路由；IPv6 报文被简单地看做 IPv4 报文内的净荷。隧道的入口点，无论是一台路由器还是主机，都要实施封装功能。在 IPv4 首部中的源 IPv4 地址带有那个节点的 IPv4 地址，目的地址是隧道端点的地址。IPv4 首部的协议字段被设置为 41（十进制），指明是一条被封装的 IPv6 报文。出口节点或隧道端点实施解封装，去掉 IPv4 首部，并依据情况，通过 IPv6 将报文路由到最终目的地。

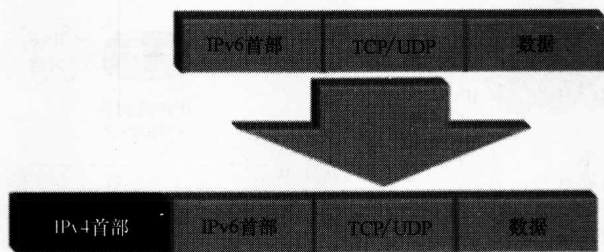


图 15-3　IPv4 隧道之上的 IPv6[11]

15.3.1　IPv4 网络上传输 IPv6 报文的打隧道场景

使用这种基本的打隧道方法，就定义了基于隧道端点的多种场景。也许最常见的配置是一种路由器到路由器的隧道，如图 15-4 所示，这是配置隧道的最常见方法。

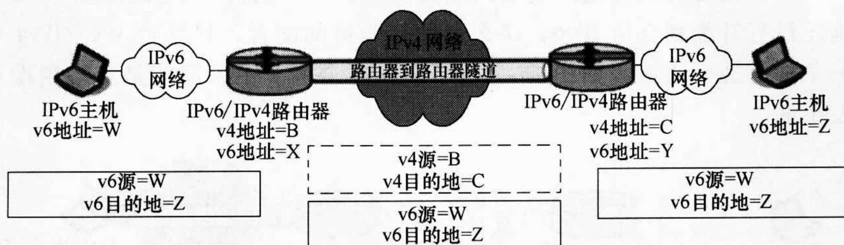


图 15-4　路由器到路由器的隧道[11]

在这个图中，在左侧的源发 IPv6 主机有 IPv6 地址 W（现在是出于简单性和简洁性考虑）。目的地为图中远端主机的一条报文㊀具有 IPv6 地址 Z，该报文被发往服务该子网的一台路由器。这台路由器，带有 IPv4 地址 B 和 IPv6 地址 X，接收到 IPv6 报文。被配置为将目的地为主机 Z 所在网络的报文以隧道方式传输，则该路由器将 IPv6 报文封装带有一个 IPv4 首部。路由器使用它的 IPv4 地址（B）作为源 IPv4 地址，以隧道端点路由器（带有 IPv4 地址 C）作为目的地地址，后者由图 15-4 中心部分 IPv4 网络之下的点线式矩形表示。被打上隧道的报文，像“常规”IPv4 报文一样被路由到目的地隧道端点路由器。这台端点路由器将报文解封装，去掉 IPv4 首部，将原始的

㊀ 这条报文被粗略地标识为图中原始主机之下的实线矩形，给出报文的 IPv6 源地址 W 和目的地地址 Z。在本图和后续的打隧道方法的图中，隧道首部被显示为点线式矩形。

IPv4 报文路由到预期的目的地 Z。

另一种隧道法场景，突出一台 IPv6/IPv4 主机，该主机能够支持 IPv4 和 IPv6 协议，以隧道方式将一条报文传输到一台路由器，接下来路由器对报文解封装，并将其通过 IPv6 以原生方式（natively）进行路由。这个流程及报文首部地址如图 15-5 所示。打隧道的机制是与路由器到路由器的情形相同的，但隧道端点是不同的。

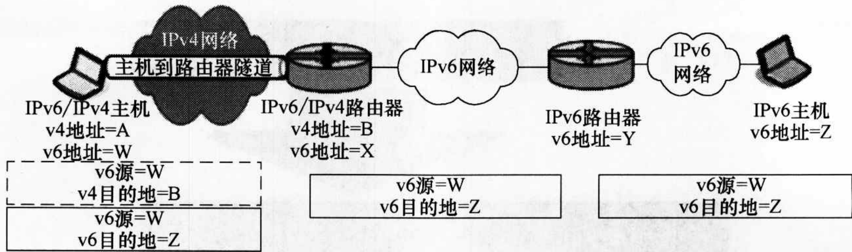


图 15-5　主机到路由器的隧道法配置[11]

路由器到主机的配置也是非常类似的，如图 15-6 所示。在图中左侧的源发 IPv6 主机，将 IPv6 报文发送到它的本地路由器，后者将报文路由到最靠近目的地的一台路由器。发挥作用（serving）的路由器被配置为在 IPv4 之上以隧道方式将 IPv6 报文传输到主机，如图 15-6 所示。

最后一种打隧道的配置是跨越端到端的配置，即从主机到主机的情形。如果路由基础设施还没有升级到支持 IPv6，那么这种打隧道的配置，使两台 IPv6/IPv4 主机能够通过一条隧道进行通信，如图 15-7 所示。在这个例子中，通信是从端到端在 IPv4 协议上发生的。

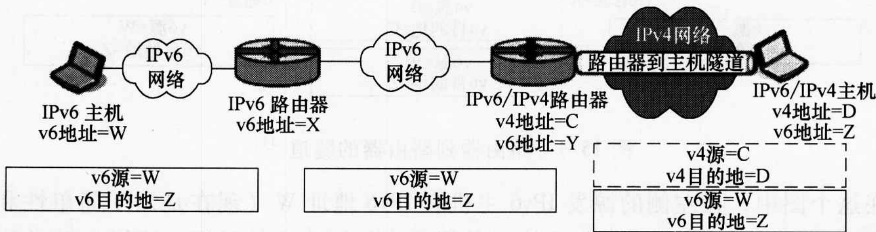


图 15-6　路由器到主机的隧道配置[11]（一）

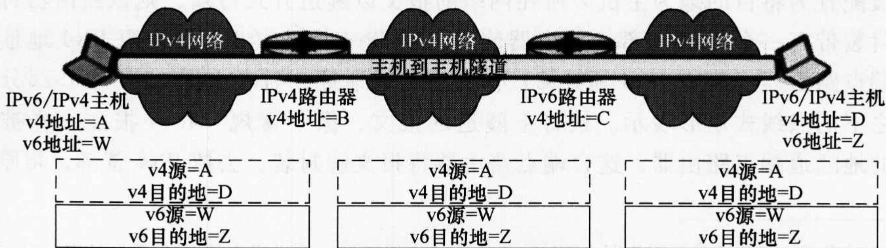


图 15-7　路由器到主机的隧道配置[11]（二）

15.3.2 隧道类型

如前面提到的，隧道是配置的或自动产生的。配置的隧道是在通信之前由管理员提前配置的。在上面描述的场景中，就何时以隧道方式传输 IPv6 报文而言，要配置设备就要求配置相应的隧道端点，即依据目的地和其他隧道配置参数进行配置，这些参数是隧道实现必备的参数。

一条自动产生的隧道并不要求隧道的提前配置，但却要求激活以隧道法传输的配置。依据 IPv6 报文内包含的信息（例如源或目的地 IP 地址）创建隧道。本节描述了如下自动方式打隧道的技术。

（1）6to4。依据一个特定的全局地址前缀和内嵌的 IPv4 地址，路由器到路由器自动打隧道方法。

（2）ISATAP。依据一种特定的 IPv6 地址格式（包括一个内嵌的 IPv4 地址），主机到路由器、路由器到主机或主机到主机的自动打隧道方法。

（3）6over4（IPv4 之上的 IPv6）。使用 IPv4 组播的主机到主机的自动打隧道方法。

（4）隧道代理。由一台服务器实现的自动隧道建立方法，在指派隧道网关资源过程中，该服务器代表要求打隧道的主机，作为一个隧道代理。

（5）Teredo。在 IPv4 网络上通过 NAT 防火墙的自动打隧道方法。

（6）双栈迁移机制。支持在 IPv6 网络上 IPv4 报文的自动打隧道方法。

6to4。6to4 是 IPv4 之上传输 IPV6 的打隧道技术，它依赖于一种特定的 IPv6 地址格式来识别 6to4 报文，并据此以隧道方式进行传输。地址格式由一个 6to4 前缀 2002::/16，后跟一个全局唯一的 IPv4 地址（用于预期的目的地站点）组成。这种串接法形成一个 48 前缀，如下图。

唯一的 IPv4 地址，在图 15-8 我们的例子中是 192.0.2.131，代表终止 6to4 隧道的 6to4 的 IPv4 地址。48bit 的 6to4 前缀用作全局路由前缀，一个子网 ID 可被附加为接下来的 16bit，后跟一个接口 ID，由此完整地定义了该 IPv6 地址。必须使用带有 6to4 隧道法支持的路由器（6to4 路由器），通过 6to4 隧道发送/接收的 IPv6 主机，必须配置有一个 6to4 地址，并被认为是 6to4 主机。

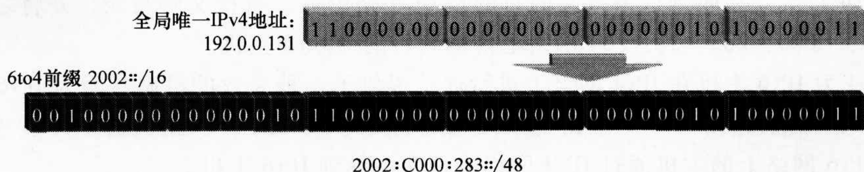


图 15-8 6to4 地址前缀生成法[147]

让我们考虑一个例子，包含 6to4 主机的两个站点，期望进行通信，并通过连接到一个共同的 IPv4 网络的 6to4 路由器进行互联；这可能是因特网或一个内部 IPv4 网络。依据图 15-9，路由器的 IPv4 接口的 IPv4 地址（相互面对）分别是 192.0.2.130 和 192.0.2.131。将这些 IPv4 地址转换到 6to4 前缀，我们分别得到 2002：C000：

282：：/48 和 2002：C000：283：：/48。就 6to4 可达性而言，这些前缀现在识别了每个站点。在左侧的我们的 6to4 主机处于子网 ID = 1 上，且出于简单性，有接口 ID = 1。因此这台主机的 6to4 地址是 2002：C000：282：1：：1。这个地址将以手工方式配置在设备上或以自动方式进行配置（自动配置基于设备的接口 ID 和路由器通告 2002：C000：282：1：：/64 前缀）。类似地，在另一站点的 6to4 主机处在子网 ID = 2 上，接口 ID = 1，得到一个 6to4 地址 2002：C000：283：2：：1。



图 15-9 6to4 隧道法的例子[147]

对应于这些 6to4 地址的 AAAA 和 PTR 资源记录也应该添加到合适域内的 DNS。当通过因特网打隧道时，目的地 AAAA 和 PTR 记录是由管理相应 6to4 设备的每个组织机构维护的，解析会要求沿每个域子树向下遍历。AAAA 记录遵循正常的“转发域”解析法，但 PTR 记录有点不是那么直接。因为 PTR 域树是基于相应 IPv6 地址的，在 6to4 情形中这种地址是组织机构依据它的 IPv4 地址空间“自配置的”，而不是由一个上游 IPv6 地址注册机构分配的，所以 ip6. apra 委派就与一个权威的上游父域没有关系的。建立了一个特殊的注册机构，处理来自 2. 0. 0. 2. ip6. arpa 区域的委派：号码资源组织（NPO）。在我们的例子中，对应于 2002：C000：283：：/48 前缀的 ipv6. arpa 域的管理员们，将带有 6to4. nro. net 的 3. 8. 2. 0. 0. 0. 0. C. 2. 0. 0. 2. ip6. arpa 区域和对应的权威名字服务器进行注册。

继续将讨论回到报文流，当在左侧的我们的主机希望与右侧的主机通信时，一次 DNS 查询将解析到它的 6to4 地址。发送主机将使用它的 6to4 地址作为源地址，使用目的地 6to4 地址作为目的地址。当 6to4 路由器接收到这条报文时，该路由器将分别使用它的（源）IPv4 地址和另一个 6to4 路由器的（目的）IPv4 地址，将报文封装上一个 IPv4 首部。接收到这条报文的目的地 6to4 路由器，将报文解封装，并将之在它的 2002：C000：283：2：：/64 网络上传输到目的地 6to4 主机。

6to4 为 IPv6 主机在 IPv4 网络上进行通信提供了一种高效的机制。因为 IPv6 网络是增量式部署的，则 6to4 中继路由器（它是也支持 6to4 的 IPv6 路由器）可被用来从“纯” IPv6 网络上的主机通过 IPv4 网络将报文中继到 IPv6 主机。

但是，可施用相同的编址和打隧道的方案，6to4 路由器要求知道 6to4 中继路由器，从而可将全局单播（纯粹的（native））IPv6 地址映射到一个 6to4 地址，便于打隧道。配置这些中继路由器，有三种方式。

（1）配置到目的地纯 IPv6 网络的路由，其中以 6to4 中继路由器作为下一跳。这种情形如图 15-10 所示。

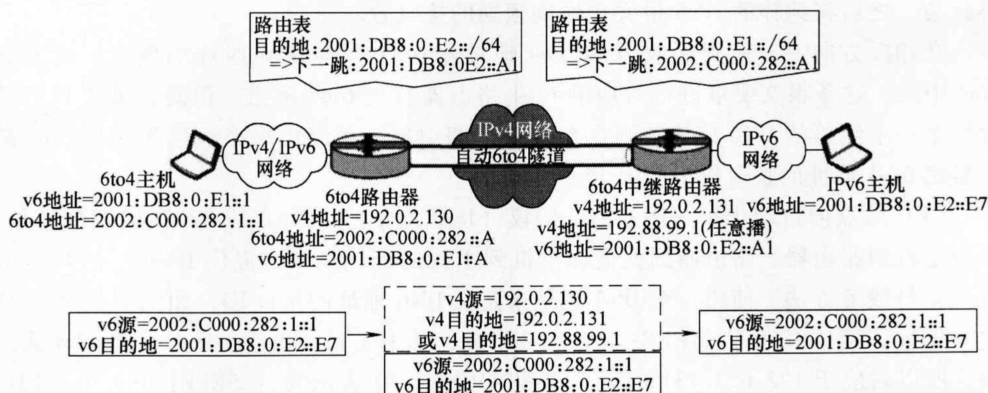


图 15-10　6to4 主机与一台纯粹的 IPv6 主机通信[147]

（2）利用正常的路由协议，使 6to4 中继路由器通告到 IPv6 网络的路由。当通告到迁移 IPv6 网络或内部的 IPv6 网络时，可应用这种场景。如果图 15-10 中的纯 IPv6 网络是“IPv6 互联网”，则图中下面的默认路由选项可能是一条较佳的可选路由。

（3）配置到 6to4 中继路由器的一条默认路由，可到达 IPv6 网络。这种场景可应用于如下情况，其中一条 IPv6 因特网连接，仅通过一个 IPv4 网络内部可达该组织机构，在该机构内不存在或几乎不存在纯 IPv6 网络㊀。

在浏览（walking through）图 15-10 时，我们在图左侧的一个 IPv4/IPv6 网络上有一台 6to4 主机，它有一个纯粹的 IPv6 地址和一个 6to4 地址。这台主机开始与图中右侧带有 IP 地址 2001：DB8：0：E2：：E7 的一台纯粹 IPv6 主机通信。当查询目的地主机的 IP 地址时，从一台 DNS 服务器返回的一条 AAAA 资源记录响应内带有这个 IPv6 地址。因此，我们在左侧的 6to4 主机，使用其 IPv6 或 6to4（显示出的）地址作为源 IP 地址（依据主机的地址选择策略）并使用目的地主机的 IPv6 地址作为目的地，形成一条 IPv6 报文。

之后这条报文到达 IPv4 网络云左侧的 6to4 路由器。为了路由到目的地 2001：DB8：0：E2：：/64 网络，该路由器需要有一条路由表项，指向 6to4 中继路由器的 6to4 地址，如图所示。之后 6to4 路由器将创建一条到相应 IPv4 地址的一条自动隧道，该 IPv4 地址被包含在路由表中可找到的 6to4 地址的第 17～48bit。注意，就像刚才讨论的，这个路由表项可以是一条默认路由，将报文路由到 6to4 中继路由器的 6to4 单播地址，也可以是 6to4 任意播地址。

虽然在图 15-10 中的路由器的路由表中没有给出，但图示在 IPv4 网络云之下的隧道式报文首部，表明封装 IPv6 报文的目的地 IPv4 地址，可以是 IPv4 单播地址或对应于 6to4 任意播地址的 IPv4 地址。在接收到 IPv4 报文时，6to4 中继路由器将对报文

㊀ 这个场景的一种变形，要求将下一跳默认路由定义为 6to4 中继路由器任意播地址（用于 IPv6 网络）。这种变形支持带有多个 6to4 中继路由器的场景。RFC 3068（194）为 6to4 中继路由器定义了一个任意播地址：2002：C058：6301：：/48。这个地址对应于 IPv4 地址 192.88.99.1。这种变形也形象地示于图 15.10 之中。

解封装，之后将纯粹的 IPv6 报文传输到预期的接收者。

在相反方向上，使用接收者的 6to4 地址作为目的地 IPv6 地址的做法，将通知 6to4 中继，这条报文要求进行对应的 6to4 路由器打上 6to4 隧道。但是，如果目的地地址是一个纯粹的 IPv6 地址，则在 6to4 中继路由器内的路由表必须包含对应 6to4 路由器之 6to4 地址的映射作为朝向 IPv6 主机的下一跳。

（1）站点内自动化打隧道的编址协议（ISATAP)。ISATAP 是一个试验型的协议，它为主机到路由器、路由器到主机和主机到主机的配置场景，提供 IPv4 之上 IPv6 的自动化打隧道方法。使用一个 IPv4 地址来定义 IPv6 地址的接口 ID，如此形成 ISATAP IPv6 地址。接口 ID 由：：5EFE：a. b. c. d 组成，其中 a. b. c. d 是点分十进制 IPv4 表示法。所以对应于 192. 0. 2. 131 的一个 ISATAP 接口 ID 表示为：：5EFE：192. 0. 2. 131。IPv4 表示法提供了这样一个清晰的指示，即 ISATAP 地址包含一个 IPv4 地址，而并不需要将 IPv4 地址转换为十六进制。这个 ISATAP 接口 ID 可被用作一个正常的接口 ID，其中将之附加到所支持的网络前缀之后，如此定义 IPv6 地址。例如，使用上述 ISATAP 接口 ID 的链路本地 IPv6 地址是 FE80：：5EFE：192. 0. 2. 131。

要求支持 ISATAP 的各主机维护一个潜在的路由器列表（PRL)，在每台路由器通告一个 ISATAP 接口时，该表包含 IPv4 地址和关联的地址寿命定时器。ISATAP 主机通过 IPv4 之上的路由器请求（solicitation）从本地路由器出请求 ISATAP 支持信息。请求目的地需要由主机加以识别，方法是预先的人工配置、在 DNS 中查找带有主机名“isatap”的路由器或使用一个 DHCP 厂商特定选项（指明 ISATAP 路由器（可能多个）的 IPv4 地址（可能多个))。DNS 技术要求管理员们使用 isatap 主机名为 ISATAP 路由器创建资源记录。

一台 ISATAP 主机将采用一个 IPv4 首部封装 IPv6 数据报文，如图 15-11 所示，它使用的是来自 PRL 的对应于选中路由器的 IPv4 地址。

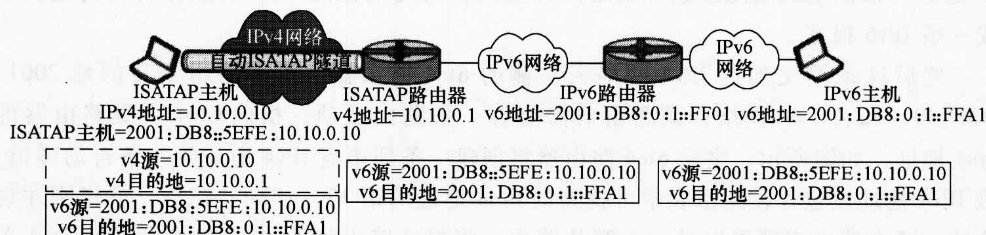


图 15-11 ISATAP 主机到路由器范例[147]

ISATAP 主机可使用配置的 IPv4 地址自动配置它们的 ISATAP 接口 ID，IPv4 地址可能是静态确定的或通过 DHCP 得到的。如果配置有 IPv6，微软 XP 和 2003 服务器可实施这样的自动配置。微软 Vista 和微软 7 客户端以及 Windows 2008 服务器默认地支持 ISATAP 自动配置。ISATAP 接口 ID 被附加在一个 64bit 全球网络前缀之后，子网 ID 是由被请求 ISATAP 路由器在其路由器通告中提供的。

按照图 15-11，在图左侧的主机使用 DNS 识别目的地主机的 IP 地址，在这种情形中是一个 IPv6 地址。一条 IPv6 报文将由主机形成，它使用其 ISATAP IPv6 地址作

为报文的源地址，目的地 IPv6 主机地址作为目的地地址。这条报文是封装在一个 IPv4 首部中的，因此形成一条自动隧道。隧道源地址被设置为 ISATAP 主机的 IPv4 地址，目的地地址被设置为 ISATAP 路由器的 IPv4 地址，在 IP 首部中的协议字段被设置为十进制的 41，指明这是一条被封装的 IPv6 报文。ISATAP 路由器不必和主机位于同一个物理网络上，隧道可跨越主机和 ISATAP 路由器之间的一个通用 IPv4 网络（零跳或多跳）。ISATAP 路由器去掉 IPv4 首部，并将得到的 IPv6 报文路由到目的地主机，使用的是正常的 IPv6 路由。

目的地主机可使用源发主机的 ISATAP 地址，对源发主机做出响应。因为 ISATAP 地址包含一个全局唯一网络前缀/子网 ID，所以返回的目的地报文被路由到发挥作用（serving）的 ISATAP 路由器。在处理接口 ID 时，ISATAP 路由器可抽取目的地主机的 IPv4 地址，并以到源发主机的一个 IPv4 首部封装该 IPv6 报文。采取一种类似方式，图 15-11 右侧中纯粹 IPv6 主机可发起到 ISATAP 主机的通信。从右到左，在这种情形中，ISATAP 路由器将生成到主机的 ISATAP 隧道。

主机到主机的 ISATAP 隧道，类似于图 15-7 中给出的隧道，可由一个 IPv4 网络上驻留的 ISATAP 主机发起，其中一个链路本地（同一子网）或全局网络前缀可被添加在每台主机 ISATAP 接口 ID 之前作为前缀。在图 15-7 中，IPv6 地址 W 和 Z 将分别代表从 IPv4 地址 A 和 D 形成的 ISATAP 地址。

（2）6over4。6over4 是一种自动打隧道技术，它利用了 IPv4 组播。要求使用 IPv4 组播，6over4 被看做一条虚链路层或虚拟以太网。由于采用了虚拟链路层的观点，所以形成 IPv6 地址时使用了一个链路本地范围（FE80::/10 前缀）。一台主机的 IPv4 地址组成了其 IPv6 地址的 6over4 接口 ID 部分。例如，带有 IPv4 地址 192.0.2.85 的一台 6over4 主机将形成一个 IPv6 接口 ID::C000:255，因此形成一个 6over4 地址 FE80::C000:255。6over4 隧道可以是主机到主机、主机到路由器和路由器到主机等形式，其中相应的主机和路由器必须被配置支持 6over4。使用相应的 IPv4 组播地址，将 IPv6 报文以隧道方式打在 IPv4 首部之内。组播组的所有成员接收到打过隧道的报文，因此这类似于虚拟链路层，预期的接收者去掉 IPv4 首部并处理 IPv6 报文。只要至少有一台 IPv6 路由器也运行 6over4，它通过 IPv4 组播机制是可达的，则该路由器就被用作一个隧道端点，通过 IPv6 路由报文。

6over4 支持 IPv6 组播和单播，所以各主机可实施 IPv6 路由器和邻居发现，来定位 IPv6 路由器。当以隧道方式传输 IPv6 组播消息时，例如为了进行邻居发现，则 IPv4 目的地地址格式为 239.192.Y.Z，其中 Y 和 Z 是 IPv6 组播地址的最后两个字节。因此到所有路由器链路范围的组播地址 FF02::2 的一条 IPv6 消息，将以隧道方式被传输到 IPv4 目的地 239.192.0.2。6over4 主机使用因特网组成员协议（IGMP）将组播组成员关系通知 IPv4 路由器。

（3）隧道代理。隧道代理为 IPv4 网络之上自动打隧道法，提供了另一项技术。隧道代理管理（代理）来自双栈客户端和隧道代理服务器（它们连接到预期的 IPv6 网络）的隧道请求。尝试访问一个 IPv6 网络的双栈客户端可有选择地被定向到一个隧道代理 web 门户，要输入认证机密信息，以此授权代理服务的使用权。隧道代理也

为授权范围管理证书。客户端也为其隧道末端提供了 IPv4 地址，还有客户端的期望 FQDN、所请求 IPv6 地址的数量以及客户端是一台主机还是一台路由器。

一旦被授权，隧道代理就实施多项任务来代理隧道的创建生成。

1）指派并配置一台隧道服务器，将所选中的隧道服务器通知新的客户端。

2）依据所请求的地址数量和客户端类型（路由器或主机），向客户端指派一个 IPv6 地址或前缀。

3）在 DNS 中注册客户端 FQDN。

4）将客户端被指派的隧道服务器和相关联的隧道、IPv6 参数（包括地址/前缀和 DNS 名）通知客户端。

在图的顶部，图 15-12 形象地说明了客户端-隧道代理的交互通信，在下面显示所得到的客户端和所指派隧道服务器之间的隧道。RFC 5572[150] 批准隧道建立协议（TSP），其中促进形成了通用的隧道建立消息和组件交互通信。

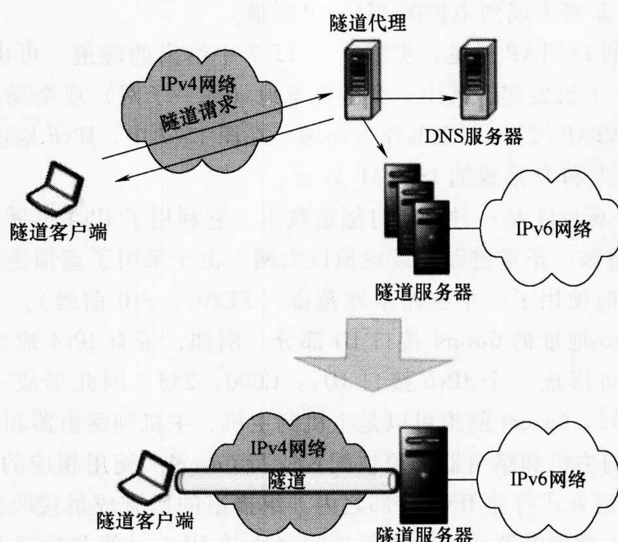


图 15-12 隧道代理交互通信[147]

（4）Teredo。通过实施网络地址转换的防火墙打隧道，如果从设计上不是不可能的话，那么也是具有挑战性的。Teredo 是一项隧道代理技术，它支持在 IPv4 之上将 IPv6 报文以 UDP 上的隧道方式进行传输而穿越 NAT，针对的是主机到主机的自动打隧道情形。为了便利 NAT/防火墙穿越，Teredo 集成了附加的 UDP 首部（见图 15-13）。许多 NAT/防火墙设备将不允许带有报文首部协议字段设置为 41（依据前面所述，这是针对 IPv6 隧道法而设的）的 IPv4 报文通过。附加的 UDP 首部进一步将隧道“埋入”（buries），以便支持它穿越通过 NAT/防火墙设备，多数这种设备均支持 UDP 端口转换。

Teredo 是在 RFC 4380[151] 中定义的，目的是提供“IPv6 接入的最无奈手段”（last resort）（这是由于它具有额外负担），但当支持 6to4 路由器或支持 IPv6 的防火墙路由器得以部署时，将用得越来越少。Tereo 要求如下元素（element），如图 15-14。

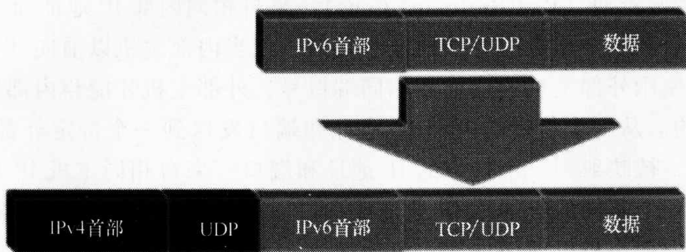


图 15-13　Teredo 隧道添加一个 UDP 首部，之后再添加 IPv4 首部[151]

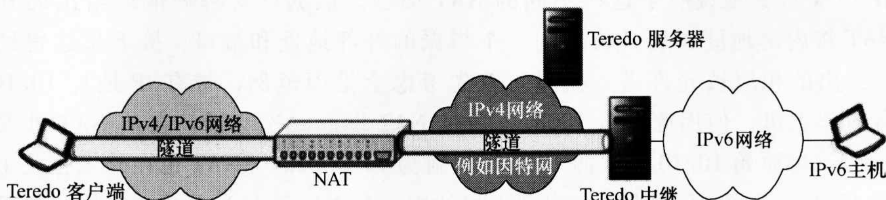


图 15-14　Teredo 客户端到 IPv6 主机的连接[147]

1）Teredo 客户端。

2）Teredo 服务器。

3）Teredo 中继。

Teredo 打隧道过程开始时，一个 Teredo 客户端实施一个资格证明过程来发现最接近预期目的地 IPv6 主机的一台 Teredo 中继，并识别正在工作的 NAT 防火墙类型。Teredo 中继是 Teredo 隧道端点，它服务预期的目的地主机。Teredo 主机必须提前配置这台 Teredo 服务器 IPv4 地址以便使用，这会帮助建立 Teredo 连接。

确定最近 Teredo 中继的过程，包括将一条 IPv6 ping（ICMPv6 echo request（回声请求））发送到目的地主机。ping 被一个 UDP 和 IPv4 首部封装，并发送到 Teredo 服务器，服务器对报文解封装，将纯粹的 ICMPv6 报文发送到目的地。目的地主机的响应将通过纯粹的 IPv6 被路由到最近的（路由角度来说）Teredo 中继，之后发回源发主机。在这种方式中，客户端依据其 IPv4 和 UDP［隧道］首部，确定合适的 Teredo 中继的 IPv4 地址和端口。图 15-14 形象地说明了这种情形，其中一个 Teredo 客户端正与一台纯粹的 IPv6 主机通信。

（5）NAT 类型。将被穿越 NAT 的类型驱动得到这样的需要，即实施一个额外步骤，使该 NAT 设备初始化 Teredo 客户端和 IPv6 主机之间数据交换的表映射。定义了如下 NAT 类型：

1）全锥面（Full cone）。来自相同内部 IP 地址和端口的所有 IP 报文，都被映射到一个相应的外部地址和端口。通过传输到被映射的外部地址和端口，外部主机是可与主机通信的。

2）受限锥面（Restricted Cone）。来自相同内部 IP 地址和端口的所有 IP 报文，都被映射到一个相应的外部地址和端口。仅当内部主机以前向外部主机发送过一条报文，一台外部主机才能与这台内部主机通信。

3）端口受限锥面（Port Restricted Cone）。来自相同内部IP地址和端口的所有IP报文，都被映射到一个相应的外部地址和端口。仅当内部主机以前向外部主机发送过一条报文，并使用外部主机的地址和相同端口号，外部主机才能与内部主机通信。

4）对称的。从一个给定的内部IP地址和端口发送到一个特定外部IP地址和端口的所有报文，被映射到一个特定的IP地址和端口。来自相同主机IP地址和端口的报文，发送到一个不同的目的地IP地址或端口，会得到一个不同的外部IP地址和端口映射。仅当内部主机以前向外部主机发送过一条报文，并使用外部主机的地址和相同端口号，外部主机才能与内部主机通信。

表15-1形象地说明了这些不同的NAT类型。依据针对每种情形给出的外发流量，NAT将内部地址和端口映射到一个指派的外部地址和端口；接下来这得到表中右侧栏给出的相应被允许进入流量。首先考虑全锥面范例，带有IP地址10.10.0.1的一台内部主机，使用源端口10081，通过NAT发起一次会话。在从NAT外发的报文上，NAT将原始10.10.0.1源IP地址映射为192.0.2.1。NAT也在外发报文上指派端口号43513，并为返回到IPv4内部主机的内部分支（流量）指派端口号42512（任何高序号的端口）。因此，NAT终止第一条连接，并将之桥接到另一条连接上；第一条连接是（10.10.0.1：10081↔NAT IP地址：42512），而第二条连接是（192.0.2.1：43513↔来自原始报文的目的地IP地址：来自原始报文的目的地端口）。

表15-1 NAT类型

外发流量					被允许的进入流量	
内部主机→外部主机			NAT映射		外部主机→内部主机	
全锥面	源	目的地	内部 ⇔	外部	源	目的地
IP地址	10.10.0.1	任意	10.10.0.1	192.0.2.1	任意	192.0.2.1
端口	10081	任意	42512	43513	任意	任意
受限锥面	源	目的地	内部 ⇔	外部	源	目的地
IP地址	10.10.0.1	203.0.113.8	10.10.0.1	192.0.2.1	203.0.113.8	192.0.2.1
端口	10081	任意	42512	43513	任意	任意
端口受限锥面	源	目的地	内部 ⇔	外部	源	目的地
IP地址	10.10.0.1	203.0.113.8	10.10.0.1	192.0.2.1	203.0.113.8	192.0.2.1
端口	10081	80	42512	43513	80	任意
对称的	源	目的地	内部 ⇔	外部	源	目的地
IP地址	10.10.0.1	203.0.113.8	10.10.0.1	192.0.2.1	203.0.113.8	192.0.2.1
端口	10081	80	42512	43513	80	43513

基于这条内部发起的公报（communique）（译者注：感觉是通信，原文错误），任一台外部主机均可发起到IP地址192.0.2.1的一条报文，由此内部连接到10.10.0.1主机。这表示为表右侧的被允许进入流量之下。在全锥面情形中，来自一台外部主机的一条进入报文，它源于任何IP地址或端口（源地址＝任意，

端口 = 任意），且目的地 IP 地址为 192.0.2.1 的任何端口上，这样的报文都将被接受。

将这种情形与表底部的对称情形比对一下，在对称情形中，一台在 IP 地址 10.10.0.1 上的主机使用源端口 10081，发起到目的地 IP 地址 203.0.113.8 端口 80 的一条连接。NAT 终止这条连接，并发起到给定目的地地址和端口的一条外部连接，将源 IP 地址映射到 192.0.2.1、端口映射到 43513。在这种情形中，被允许穿过 NAT 的仅有可接受进入流量，将具有一个源地址和端口组合 203.0.113.8：80、目的地地址和端口组合 192.0.2.1：43513，这是依据来自源发主机的通信进行映射得到的。

受限锥面配置法将原始目的地 IP 地址映射为被允许的，即当相应目的地 IP 地址是 NAT 映射的地址（在这个例子中是 192.0.2.1）时，来自外部空间的后续进入源 IP 地址（203.0.113.8）为允许的。端口受限场景加入了端口有效性检查，方法是当相应目的地 IP 地址和端口匹配 NAT 映射到地址（在这个例子中是 192.0.2.1）时，将来自外部空间（203.0.113.8：80）的后续进入 IP 地址和端口是允许的。

为了使用 Teredo 进行通信并建立防火墙穿越，这种 NAT 类型驱动 Teredo 初始化过程。当是全锥面时，NAT 的识别不要求进一步的合格性检查，原因是任何外部主机可发起到其外部地址的通信。但无论哪种受限锥面场景都要求进一步的合格性检查，以便合适地将地址和目的地地址映射到 NAT 内的那些对应的地址和目的地。为了完成 NAT 内部主机与目的地主机通信的映射，Teredo 客户端将一条冒泡（bubble）报文发送到目的地主机。一个冒泡报文是没有净荷的一个 IPv6 首部，它被封装在 Teredo 隧道 IPv4/UDP 首部之中。针对端口受限锥面场景，它使 NAT 能够完成内部和外部 IP 地址、内部和外部端口号的映射。

一般而言，冒泡报文是从源 Teredo 客户端直接发送到目的地主机的。但如果目的地主机也位于一个防火墙之后，那么就丢弃冒泡报文，原因是这是一条未被请求（unsolicited）的外部报文。在这种情形中，Teredo 客户端超时，并通过 Teredo 服务器发送冒泡报文，该服务器是由预期目的地 Teredo 格式的 IPv6 地址加以识别的，由该地址对 Teredo IPv4 地址编码。

假定目的地主机也是一台 Teredo 客户端，它将接收该报文，它已经在客户端配置过程中，由一条以前发送到这台 Teredo 服务器的 ping 进行了初始化。之后目的地主机将直接对源发主机做出响应，这就完成了 NAT 映射（在两端都是如此）。图 15-15形象地说明了这个场景，其中两台 Teredo 客户端通过一个共同的 Teredo 中继进行通信。Teredo 不支持对称 NAT 设备的自动穿越。

如我们看到的，Teredo IPv6 地址的格式带有客户端及其服务器 Teredo 服务器的 IPv4 地址。Teredo IPv6 地址的格式如图 15-16 所示。

Teredo 前缀是一个预先定义的 IPv6 前缀：2001::/32。Teredo 服务器的 IPv4 地址组成了接下来的 32bit。各标志指明 NAT 类型为全锥面（十六进制 =8000）或受限的或端口受限的（十六进制 =0000）。客户端端口和客户端 IPv4 地址字段表示这些对应值混杂后的值，采取的方法是将每个比特值取反。

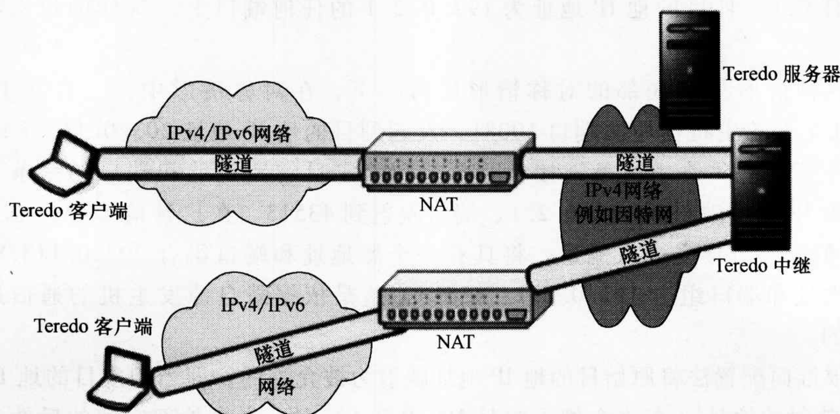


图 15-15　通过 IPv4 因特网进行通信的两台 Teredo 客户端[147]

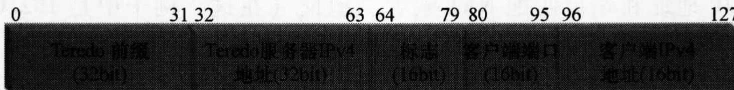


图 15-16　Teredo IPv6 地址格式[151]

15.3.3　IPv6 网络上传输 IPv4 报文的打隧道场景

在一个 IPv6 实现过程中，在 IPv6 网络上的一些 IPv6 客户端可能仍然需要与 IPv4 网络（例如因特网）上的 IPv4 应用或主机进行通信。在 IPv6 网络上对 IPv4 报文打隧道，就为保留这条通信路径提供了一种方法。

（1）双栈迁移机制（DSTM）。DSTM 提供了在 IPv6 网络上对 IPv4 报文打隧道的一种方法，报文最终是发送到目的地 IPv4 网络和主机的。打算与 IPv4 主机通信的 IPv6 网络上的主机，将要求具备双栈，同时是一个 DSTM 客户端。在使用 DNS 将预期目的地主机的主机名仅解析到一个 IPv4 地址时，客户端将发起 DSTM 过程，这非常类似于隧道代理方法。过程开始时，DSTM 联系一台 DSTM 服务器，以便得到一个 IPv4 地址（这里首选通过 DHCPv6 协议㊀）和 DSTM 网关的 IPv6 地址。IPv4 地址被用作将被传输的数据报文的源地址。这条报文被封装一个 IPv6 首部，使用的是 DSTM 客户端的源 IPv4 地址，DSTM 网关的 IPv6 地址作为目的地址。IPv6 首部中的下一首部字段指明采用这种“4over6”隧道方法的一条被封装 IPv4 报文。

DSTM 的一个变形支持纯粹（native）网络之外一台 DSTM 客户端（例如一名在家工作的工作人员（home-based worker））基于 VPN 法的访问。在这个场景中，假定 DSTM 客户端得到一个 IPv6 地址，但得不到 IPv4 地址，它就能够连接到该 DSTM 服务器来得到一个 IPv4 地址。这种访问方法要求认证，才能建立 DSTM 客户端和 DSTM 网关之间的一条 VPN。

㊀　虽然 DSTM RFC 草案[152]指明 DHCPv6 作为得到一个 IPv4 地址的首选方法，但 DHCPv6 目前还没有定义以本地方法或通过一个选项设置而指派 IPv4 地址的过程或方法。

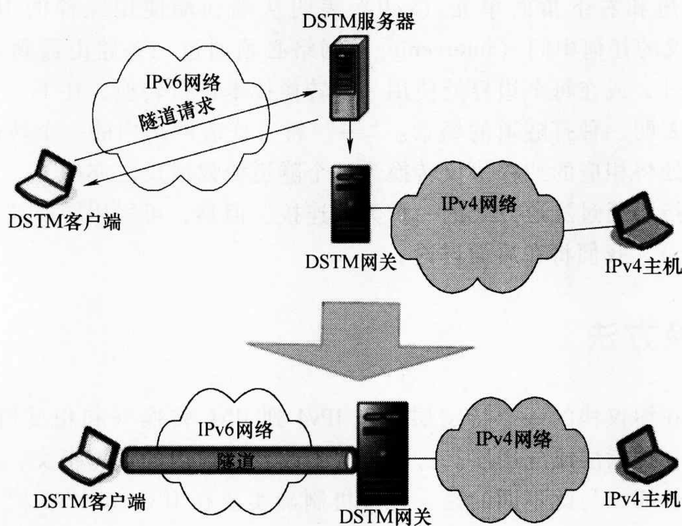


图 15-17 DSTM 隧道建立[147]

15.3.4 隧道法总结

依据源主机能力/网络类型以及目的地地址解析和网络类型，下表汇总了打隧道方法的可施用性和范围（applicability）。

从(from)	到(to)					
	IPv4 网络上的 IPv4 目的地	在 IPv4 网络上的双栈目的地，解析到 IPv4 地址	IPv4 网络上的双栈目的地，解析到 IPv6 地址	IPv6 网络上的双栈目的地，解析为 IPv4 地址	IPv6 网络上的双栈目的地，解析为 IPv6 地址	IPv6 网络上的 IPv6 目的地
IPv4 网络上的 IPv4 客户端	纯粹的 IPv4	纯粹的 IPv4		纯粹的 IPv4→IPv4 兼容的		
IPv4 网络上的双栈客户端	纯粹的 IPv4	纯粹的 IPv4	主机到主机，IPv4 之上的 IPv6①	纯粹的 IPv4→IPv4 兼容的	主机到路由器，IPv4 之上的 IPv6①	主机到路由器，IPv4 之上的 IPv6①
IPv6 网络上的双栈客户端	DSTM→纯粹的 IPv4	DSTM→纯粹的 IPv4	纯粹的 IPv6→路由器主机，IPv4 之上的 IPv6①	DSTM	纯粹的 IPv6	纯粹的 IPv6
IPv6 网络上的 IPv6 客户端			纯粹的 IPv6→IPv4 之上的 IPv6①		纯粹的 IPv6	纯粹的 IPv6

① 解析到一个 IPv6 地址的可能是一个纯粹的 IPv6 地址，或一个 6to4、ISATAP、Teredo、6over4 或 IPv4 兼容的地址。主机必须依据它对相应技术的支持，来选择目的地地址。

在表左上角和右下角的单元（cell）表明从端到端使用纯粹的 IP 版本。不同（opposite）协议的任何中间（intervening）网络必须通过一条路由器到路由器的隧道以隧道方式穿过，或在每个边界处使用一种转换技术进行转换，在下一节讨论。

其他单元表明一种打隧道的场景。"→"符号代表网络内的一个转换点或打隧道的端点，该点处将相应的纯粹协议转换为一个隧道协议或反之亦然。

空白单元指明通过隧道方式的一种无效连接。但是，可采用转换技术来桥接这些连接间隙（gap），我们将在后面讨论。

15.4 转换方法

转换技术在协议栈的一个特定层实施 IPv4 到 IPv6 转换（和相反情况），典型情况下有网络层、传输层或应用层。打隧道法不改变隧道内的数据报文，仅仅是附加一个首部或两个首部，与此不同的是，转换机制确实要在 IPv4 和 IPv6 相互之间修改或转换 IP 报文。一般而言，在仅支持 IPv6 的节点与仅支持 IPv4 的节点之间通信的一个环境中，建议使用转换方法；即，对于上面汇总表中空白单元的场景采用这种方法。在双栈环境中，首选纯粹的或打隧道机制㊀。

15.4.1 无状态 IP/ICMP 转换算法

转换 IPv4 和 IPv6 报文的常用算法是无状态的 IP/ICMP 转换（SIIT）算法。SIIT 提供了 IPv4 和 IPv6 之间 IP 报文首部的转换。SIIT 驻留在一台 IPv6 主机或网关上，将外发 IPv6 报文首部转换为 IPv4 首部，将进入的 IPv4 首部转换为 IPv6 首部。为了实施这项任务，必须向 IPv6 主机提供一个 IPv4 地址，可以是配置在该主机上的，或通过在 RFC 2765[153] 中未指派的一项网络服务得到的。当 IPv6 主机期望与一台 IPv4 主机通信时，SIIT 算法将 IPv6 报文首部转换为 IPv4 首部格式。SIIT 算法识别这样一种情形，其中当 IPv6 地址是一个 IPv4 映射的地址时的情况，其格式如图 15-18 所示。将所解析 IPv4 地址转换为一个 IPv4 映射地址的机制，由协议栈肿块（bump-in-the-stack，BIS）或 API 肿块（bump-in-the-API，BIA）技术加以提供，后面描述这些技术。

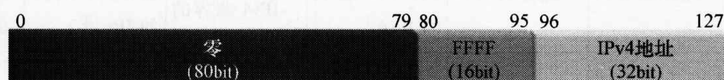


图 15-18 IPv4 映射的地址格式[12]

依据存在 IPv4 映射地址格式作为目的地 IP 地址的情况，SIIT 实施首部转换（下面描述），来得到通过数据链路和物理层进行传输的一条 IPv4 报文。一个 IPv6 节点的源 IP 地址使用一个不同的格式，即 IPv4 转换的格式，如图 15-19 所示，但 RFC 2765 却没有规范这个地址最初是如何配置的。

㊀ 依据的是 RFC 2766[156]

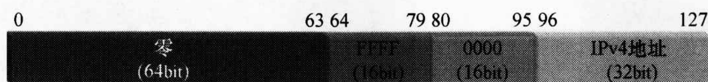


图 15-19 在 SIIT 内使用的 IPv4 转换的地址格式

SIIT 算法的一种潜在可能的协议栈视图如图 15-20 所示。

两个方向的基本首部转换过程汇总如下。

IPv4→IPv6 首部转换	IPv6→IPv4 首部转换
版本 =6	版本 =4
流量类 = IPv4 首部 TOS 比特	首部长度 =5(没有 IPv4 选项)
流标签 =0	服务类型 = IPv6 首部流量类字段
净荷长度 = IPv4 首部总长度值 -(IPv4 首部长度 + IPv4 选项长度)	总长度 = IPv6 首部净荷长度字段 + IPv4 首部长度
	标识 =0
下一个首部 = IPv4 首部协议字段值	标志 = 不分段 =1,更多分段 =0
跳限制 = IPv4 TTL 字段值 -1	分段偏移 =0
源 IP 地址 =0:0:0:0:FFFF::/96 与 IPv4 首部源 IP 地址串接	TTL = IPv6 跳限制字段值 -1
	协议 = IPv6 下一个首部字段
目的地 IP 地址 =0:0:0:0:FFFF::/96 与 IPv4 首部目的地 IP 地址串接	首部校验和 = 在 IPv4 首部上计算得到的
	源 IP 地址 = IPv6 源 IP 地址字段(IPv4 转换的地址)的低 32bit 目的地 IP 地址 = IPv6 目的地 IP 地址字段(IPv4 映射的地址)的低 32bit 选项 = 无

现在让我们看看采用 SIIT 算法转换 IPv4 和 IPv6 报文的一些技术。

15.4.2 协议栈中的肿块

协议栈中的肿块（BIS）[154]使主机利用 IPv4 应用在 IPv6 网络上进行通信。BIS 嗅探 TCP/IPv4 模块和链路层设备（例如网络接口卡）之间的数据流，并将 IPv4 报文转换到 IPv6。BIS 的各组件如图 15-21 所示。

依据 SIIT 算法，转换器组件将 IPv4 首部转换到一个 IPv6 首部。扩展名字解析器嗅探 A 记录类型的 DNS 查询；在检测到这样的一条查询时，扩展名字解析器组件就针对同一主机域名（Qname）和类（Qclass），产生查找 AAAA 记录类型的一条附加查询。如果从 AAAA 查询没有接收到肯定的应答，那么通信接下来使用 IPv4；如果 AAAA 查询被成功地解析，那么扩展名字解析器就指令地址映射器组件将返回的 IPv4 地址（A 记录）与返回的 IPv6 地址（AAAA 记录）关联。如果仅接收到一条 AAAA 响应，那么地址映射器就从一个内部配置的地址池中指派一个 IPv4 地址。

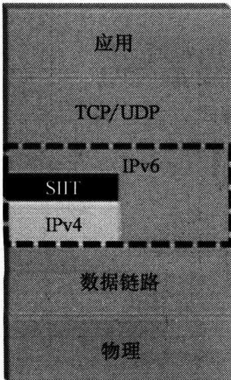


图 15-20 SIIT 栈范例[153]

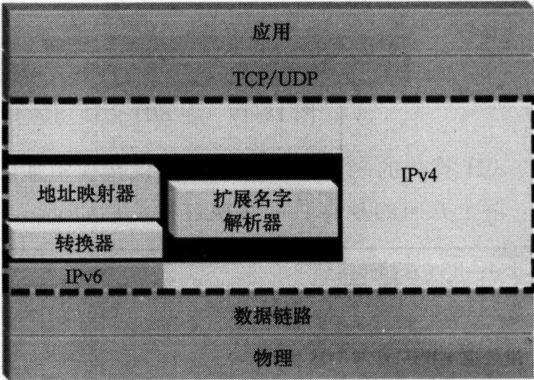


图 15-21 协议栈中肿块（BIS）组件[154]

为了沿协议栈向请求对 A 查询解析的应用提供一个响应，需要 IPv4 地址。因此，地址映射器维护真实或自指派的 IPv4 地址与目的地 IPv6 地址的关联关系。之后目的地为那个 IPv4 地址的任何数据报文，由转换器转换为 IPv6 报文，以便通过 IPv6 网络进行传输。

在 BIS 主机接收到从一台外部主机（还没有被映射）发起的一条 IPv6 报文的情形中，地址映射器将从其内部地址池指派一个 IPv4 地址，并将 IPv6 首部转换为 IPv4 首部，以便沿协议栈向上传递。

15.4.3 API 中的肿块

API 中的肿块（BIA）[155]策略支持使用 IPv4 应用，同时在一个 IPv6 网络上通信。不像 BIS 提供的 IP 首部修改的是，BIA 方法在 IPv4 和 IPv6 API 之间进行转换。BIA 是在主机的应用和协议栈的 TCP/UDP 层之间实现的，它由一个 API 转换器、一个地址映射器、名字解析器和功能映射器组成，如图 15-22 所示。

当 IPv4 应用发送一条 DNS 查询，以便确定一个目的地主机的 IP 地址时，名字解析器截获该查询，并生成一条请求 AAAA 记录的附加查询。带有一条 A 记录的一个 DNS 应答，将提供带有给定 IPv4 地址的答案。仅带有一个 AAAA 记录的一个应答，会促使名字解析器从地址映射器处请求一个 IPv4 地址，以便映射到返回的 IPv6 地址。名字解析器利用被映射的 IPv4 地址，向应用返回一条 A 记录响应。地址映射器维护 IPv6 地址到那些从一个内部地址池（由未指派 IPv4 地址空间（0.0.0.0/24）组成）指派地址间的这个映射。功能映射器截获 API 功能调用，并将 IPv4 API 调用映射到 IPv6 套接字调用。

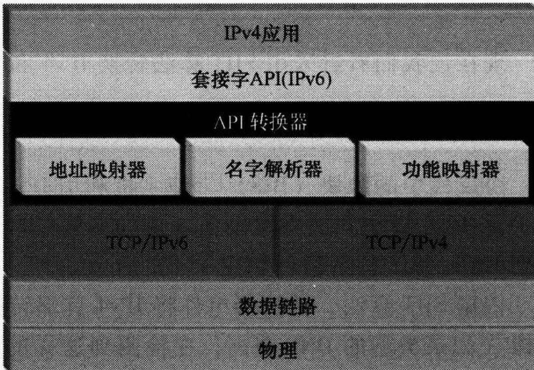


图 15-22 API 中的肿块（BIA）[155]

15.4.4 带有协议转换的网络地址转换（NAT-PT）——被废弃不用

正如其名字所蕴含的，NAT-PT[156]过程不仅涉及将 IPv4 地址转换为 IPv6 地址（就像人们所熟悉的 IPv4 NAT 那样），而且实施协议首部转换，这点见 SIIT 一节所描述的内容⊖。一台 NAT-PT 设备用作一个 IPv6 网络和一个 IPv4 网络之间的网关，并支持纯粹的 IPv6 设备与 IPv4 因特网上的主机进行通信（比如）。NAT-PT 设备维护一个 IPv4 地址池，并将一个给定 IPv4 地址与一个 IPv6 地址关联，而通信继续进行。图15-23形象地说明了一种 NAT-PT 部署的架构。由于 RFC 4966[157]中枚举到的多项原因，NAT-PT（和下面描述的 NAPT-PT）已被废弃不用，且不应进行部署。

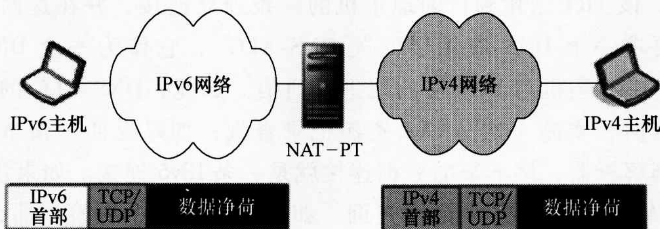


图 15-23 NAT-PT 部署（被废弃不用）[147]

15.4.5 带有协议转换的网络地址端口转换（NAPT-PT）——废弃不用

NAPT-PT 使 IPv6 节点能够使用单一 IPv4 地址与 IPv4 节点通信。因此，在上面的图 15-23 中，它并不像 NAT-PT 中那样维护一个 IPv6 地址和一个唯一 IPv4 地址的一到一关联，NAPT-PT 将每个 IPv6 地址映射到一个共同的 IPv4 地址，在相应的 IPv4 报文中设置一个唯一的 TCP 或 UDP 端口值。使用单一共享 IPv4 地址的做法，最小化了 NAT-PT 场景中 IPv4 地址池耗尽的可能。

15.4.6 SOCKS IPv6/IPv4 网关

在 RFC 1928[158]中定义的 SOCKS，为应用穿越防火墙提供了传输中继，这实际上提供了应用代理服务。针对转换 IPv4 和 IPv6 通信，RFC 3089[159]施用 SOCKS 协议。就像前面已经讨论的其他转换技术一样，这种方法包括特殊的 DNS 处理，称之为 DNS 名字解析委派，它将来自解析器客户端的名字解析任务委派给 SOCKS IPv6/IPv4 网关。一个 IPv4 或 IPv6 应用可被"套接字化"（socksified），以便为到支持其他（opposite）协议的一台主机的最终连接，而与 SOCKS 网关代理通信。图 15-24 形象地说明了一台 IPv6 主机配置有一个 SOCKS 客户端的情形，该客户端连接到一个 IPv4 主机。一台套接字化的 IPv4 主机仅仅通过该 SOCKS 网关而与一台 IPv6 主机通信，图中的方向是从右到左。

⊖ 例外情况是，在 NAT－PT 网关内由关联所控制的源和目的地 IP 地址字段。

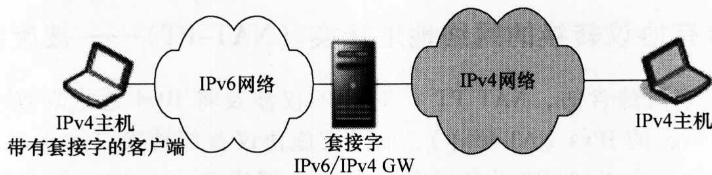


图 15-24　基本的 SOCKS 网关配置[159]

15.4.7　传输中继转换器

非常像SOCKS配置情况的是，传输中继转换器（TRT）的特征是，一台有状态网关设备互连在不同网络上的两条“独立的”连接。来自一台主机的 TCP/UDP 连接在 TRT 上终结，该 TRT 创建到目的地主机的一条独立连接，并在这两条连接之间进行中继。TRT 要求一个 DNS-应用层网关 DNS-ALG㊀，它作为一个 DNS 代理。规范 TRT，用来支持 IPv6 主机与 IPv4 目的地进行通信。如此，DNS-ALG 的主要功能是当 IPv6 解析器请求时，实施一次 AAAA 资源记录查询；如果返回一条 AAAA 记录，则应答就被传递到解析器，接下来的数据连接就是一条 IPv6 连接。如果没有返回 AAAA 记录，则 DNS-ALG 实施一次 A 记录查询，如果接收到一个应答，那么 DNS-ALG 是使用包含于所返回 A 记录中的 IPv4 地址，形成一个 IPv6 地址。RFC 3142[160]，它将 TRT 定义为一个信息型的 RFC，规定使用前缀 C6::/64 后跟 32 个零，再加上 32bit 的 IPv4 地址。但是，IANA 并没有分配 C6::/64 前缀。因此，那么就要求使用一个本地配置的前缀。

15.4.8　应用层网关

类似于 HTTP 代理，应用层网关（ALG）在应用层实施协议转换，并实施应用代理功能（见图 15-25）。典型情况下，一个客户端的应用将需要配置有代理服务器的 IP 地址，在打开应用（例如 HTTP 代理情形的网页浏览器）时，将形成到该代理服务器的一条连接。在一个仅支持 IPv6 网络上的各主机，对于到 IPv4 互联网的网页访问或其他应用特定的访问，使用一台 ALG 就是有用的。

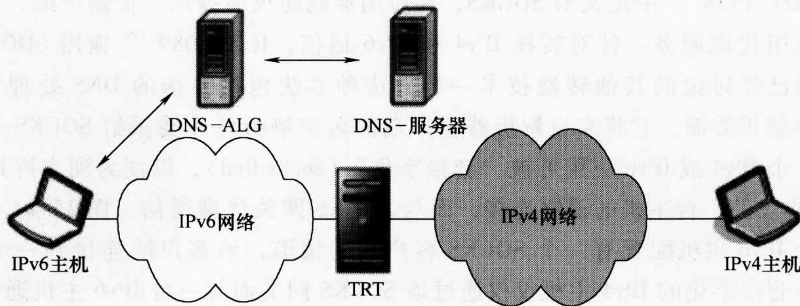


图 15-25　带有 DNS-ALG 的 TRT 配置[160]

㊀ 有时被称作“不给就捣蛋 DNS-ALG”或 totd。

15.5 应用迁移

TCP/IP 应用事实上的应用编程接口（API）是套接字接口，它最初是在 BSD UNIX 上实现的（BIND 也是最初在这个平台上实现的）。套接字接口定义了程序调用，支持应用与 TCP/IP 层接口，以便在 IP 网络上通信。微软的 Winsock API 也是基于套接字接口的。为了支持 IPv6 的较长地址池和附加特征，套接字和 Winsock 接口都做了修改。事实上，多数主要的操作系统都实现了对套接字或 Winsock 的支持，包括微软（XP SP1、Vista、7、Server 2003 和 2008）、Solaris（8+）、Linux（内核 2.4+）、Mac OS（X.10.2）、AIX（4.3+）和 HP-UX（带有升级的 11i）。更新后的套接字接口支持 IPv4 和 IPv6，并提供 IPv6 应用与 IPv4 应用使用 IPv4 映射的 IPv6 地址进行互操作的能力。和您的应用厂商核对一下对 IPv6 兼容性及其需求。

15.6 规划 IPv6 部署过程

15.6.1 服务提供商部署选项

服务提供商，特别是驻地宽带服务提供商，可在他们的网络内实现 IPv6，并最终将 IPv6 地址部署到客户处。除了应用本章讨论的各项技术外，已经形成另外两项可选技术。

1）6rd。IPv6 快速部署定义了对 6over4 迁移方法的一个变种，支持向端客户提供 IPv6 地址，同时继续支持一个 IPv4 和 IPv6 基础设施。

2）小型的双栈（Dual-Stack Lite）。为服务提供商提供了较佳利用日渐稀缺 IPv4 地址而同时将 IPv6 部署到客户端的一种方法。

（1）6rd（IPv6 快速部署）。RFC 5569[161] 定义了“在 IPv4 基础上的 IPv6 快速部署（6rd）”，这样一种技术使一个服务提供商向端客户提供 IPv6 地址，而同时维持一个 IPv4 基础设施。这种方法要求从客户端到一个 IPv6 目的地，通过修改的 6to4 技术，将客户的 IPv6 流量打上隧道进行传输。修改涉及使用服务提供商的 IPv6 前缀（/32），来替代 6to4 前缀 2001::/16。

就像 6to4 一样，IPv6 前缀接下来的 32bit 由 6to4 网关的 IPv4 地址组成，在这种情形中网关是客户端宽带路由器。因此，一个 6to4 前缀定义为 2001:<32bit IPv4 地址>::/48，而 6rd 前缀是<32bit 服务提供商 IPv6 前缀>:<32bit IPv4 地址>::/64[⊖]。这使服务提供商向每个客户提供一个/64 前缀，这组成单一 IPv6 子网。因此，带有一个 RIR 分配的 IPv6 地址块 2001:db8::/32 的一个服务提供商，将向有 IPv4 地址 192.0.2.130 的一台客户网关设备，提供一个 6rd 子网地址 2001:db8:c000:282::/64，如图 15-26 所示。

⊖ 如果人们希望的话，可使用 IPv4 地址的一部分，比如一个常用地址 10.0.0.0 的低 24 个唯一 bit，这允许较长的服务提供商前缀。

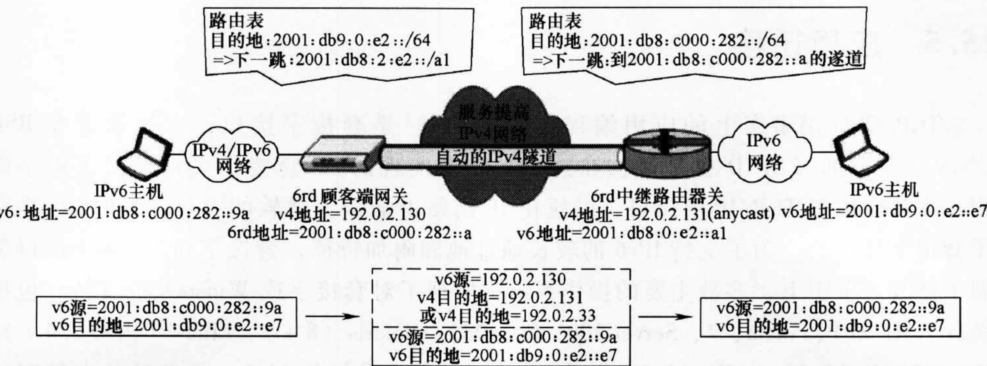


图 15-26　6rd 部署范例

在驻地（residence）内要求一个 IPv6 地址的一台设备，将从这个子网内被指派一个地址。例如在图 15-26 中，一台 PC 被指派 IPv6 地址 2001：db8：c000：232：：9a。6rd 客户网关将纯粹的 IPv6 报文在 IPv4 上打上隧道传输到一个 6rd 网关。在 6rd 和 6to4 之间的另一项地址相关的变化，是任意播 6to4 地址是固定的（192.88.99.1），而 6rd 任意播地址是由服务提供商自己在其自己的地址空间内确定的。必须向每台客户路由器提供 6rd 中继代理或任意播地址（可能多个）。

6rd 中继路由器终结 IPv4 隧道，之后将 IPv6 报文以纯粹方式（IPv6）传输到它的目的地。服务提供商前缀的使用，使 6rd 可达的目的地与服务提供商纯粹 IPv6 流量一起被通告传输。

小型双栈（Dual-Stack Lite）。小型双栈是这样一种技术，它使一个服务提供商能够更有效地利用正在消失的可用公开 IPv4 地址池，同时便于对指派给客户网关设备 IPv4 地址的长期支持[162]。典型情况下，服务提供商将一个地址指派给一台客户路由器或网关，该设备直接与宽带接入网络接口。在将 IP 地址指派给家乡网络中的 IP 设备时，客户网关实施 DHCP 服务器功能。人们预料，这样的家乡网关设备将在相当长时间内仅支持 IPv4。

组成一个小型双栈实现的组件包括如下单元。

1）基本桥接宽带（B4）单元，它将 IPv4 家乡网络与一个 IPv6 网络桥接在一起；B4 功能可驻留在客户网关设备上或在服务提供商网络内。

2）B4 和 AFTR 之间软件实现的 IPv6 中的 IPv4 隧道。

3）地址族转换路由器（AFTR）以 B4 单元终结 IPv6 中传输 IPv4 的软件隧道，它也实施 IPv4-IPv4 网络地址转换功能。

图 15-27 形象地说明了在一条端到端连接内这三个组件间的相互关系。从图的左侧开始，IPv4 主机从客户网关的 DHCP 服务器功能得到一个 IPv4 地址 10.1.0.2。让我们假定这个 IPv4 主机希望连接到一个网站，该网站已经解析到 IP 地址 192.0.2.21。IPv4 主机以源地址 10.1.0.2 和源端口 1000（比如）以及目的地址 192.0.2.21 端口 80，形成一条 IP 报文。该主机将这条报文传输到它的默认路由，即客户网关。

在这个例子中的客户网关包括 B4 单元，如果隧道还没有建立的话，那么该单元建立软件 IPv6 中的 IPv4 隧道。在客户网关的 WAN 接口（面向服务提供商网络）上已经指派一个 IPv6 地址，隧道是在这条连接上建立的。已经由人工或通过 DHCPv6 为客户网关配置了 AFTR IPv6 地址。如图 15-27 所示，B4 单元将原始 IPv4 报文封装上一个 IPv6 首部，并将报文传输到 AFTR。

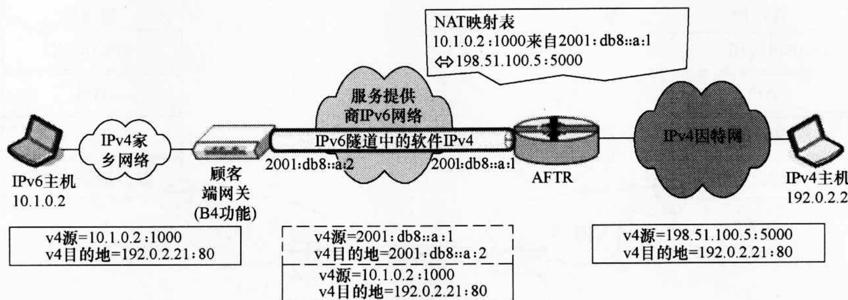


图 15-27　小型双栈架构[162]

AFTR 终止隧道，并去掉 IPv6 首部。之后 AFTR 实施一项 IPv4-IPv4 NAT 功能。为了将原始报文的私有（RFC 1918）IPv4 源地址转换为一个公开的 IPv4 地址，要求这样做。因此，服务提供商必须准备一个公开 IPv4 地址池，可被用作目的地为一个 IPv4 目的地的报文源 IP 地址，就像这种情况中那样。这种准备地址池的做法，使服务提供商能够更加高效地利用日益稀缺的公开 IPv4 地址空间。一般而言，AFTR 也实施端口转换，并为了在两个方向上正确地映射 IPv4 地址和端口号，必须跟踪每项 NAT 操作的这种映射。

在图 15-27 中，AFTR 将客户的源 IPv4 地址和端口 10.1.0.2：1000 映射到 192.51.100.5：5000。因为所有客户将利用 10.0.0.0 地址空间，所以 NAT 映射表也跟踪报文源自其上的隧道。最终被传输到目的地主机的报文，包括这个被映射的 IPv4 地址和端口 198.51.100.5：5000。目的地为这个地址/端口的返回报文，被映射到 [目的地] 地址 10.1.0.2：1000，并以隧道方式传输到 2001：db8：：a：1。

部署了纯粹 IPv6 的客户或双栈主机，可拥有 DHCPv6 功能提供的或通过自动配置得到的相应 IPv6 地址，DHCPv6 功能是在客户网关中实现的。在家乡网络上被传输到客户网关的 IPv6 报文，不会利用软件隧道，但相反，它是以纯粹的方式（IPv6）在服务提供商 IPv6 接入网络上被路由的。

15.6.2　企业部署场景

当考虑一种 IPv6 实现方法时，当然不会缺少技术选择。具有多种选择是不错的，但它也可能令人感到恐惧。选择正确的途径将取决于您的当前环境，包括端用户设备和操作系统、路由器模型和版本，以及关键应用、预算和资源，还有时间窗（time frame）（即选择合适的时机）。考虑到主导操作系统和联网设备中双栈支持的迅速增长，双栈方法可能是最普遍采用的方法。在本节，我们将回顾一下基本 IPv6 实现场

景，来提供各种宏观层次方法的一种感觉（flavor）。在回顾这些场景时，让我们使用图 15-28 中的基本图作为一个基线。在这个图中，我们有一个客户端，在这种情形中带有 IPv4 应用、IPv4 套接字 API 和 TCP/IPv4 协议栈，还有具备相当配置的一台服务器。我们将互联网络分割成分别用于客户端和服务器的接入网络以及一个核心或骨干网络。

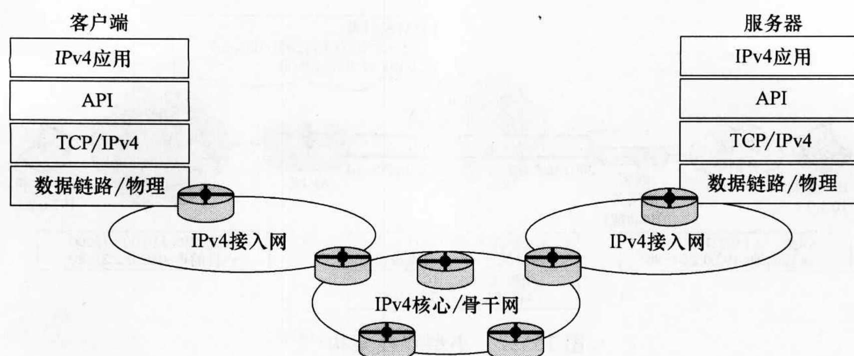


图 15-28 IPv4 网络的基本情形——迁移前的初始状态[147]

注意这个基本图形象地说明了成对的客户端-服务器连接。对于一个给定用例，这可能代表一台内部客户端通过完全处于内网的接入网络和核心网络，访问一台内部服务器的情形。但也可能代表一台内部客户端和内部服务器在因特网上通信，一台内部客户端与一台基于因特网的服务器通信，或一台外部客户端通过因特网与一台内部服务器通信等情形。这种惯例（convention）简化了独特用例的庞大数量。当描述迁移场景时，在合适的地方将指出这些变形间的微小差异。

15.6.3 核心网络迁移场景

第一个场景涉及最初为骨干或核心网络增补 IPv6 支持。这个场景要求升级所有的核心路由器来支持 IPv6 路由和路由协议，升级接入到核心的边界路由器来支持双栈。核心网络可能是一个内部骨干或一个 IPv6 ISP 网络。接入-核心边界路由器的一种常见实现方法是在它们之间利用配置的隧道。这种做法使这些边界路由器能够通告 IPv4 路由，并在 IPv6 骨干上以隧道方式传输 IPv4 报文。另外，在这些边界点上可能使用转换网关。无论采取哪种方式，这种正确实现的方法应该对客户端或服务器设备或软件几乎没有多少影响，并可在不影响端用户的条件下，为 IPv6 试验提供一个起点（见图 15-29）。

15.6.4 服务器侧迁移

接下来的场景假定我们的基本情形作为初始状态，后跟将服务器和应用主机升级到双栈实现。在服务器仍然能够支持 IPv4 通信和应用的前提下，端客户端应该像以前一样通过 IPv4 通信。但是，服务器也能够服务 IPv6 客户端。这个场景可能反映了如下用例：

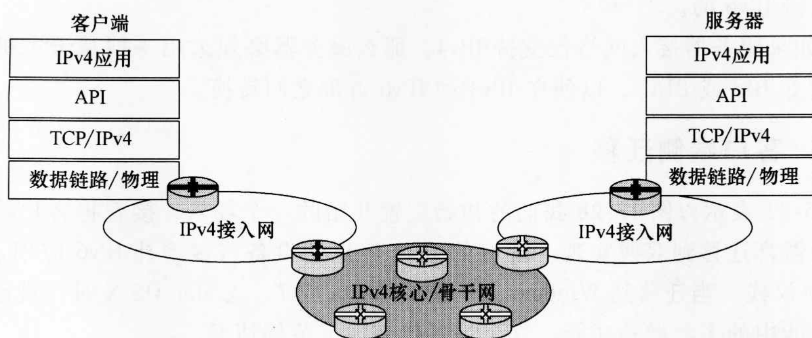


图 15-29　核心（网络）迁移场景[147]

（1）图 15-30 中的客户端和服务器处在同一个组织机构内，该组织机构能够仅升级它的服务器，以便在升级和影响端客户端之前，为 IPv6 提供整体准备性测试。在这种情形中，端客户端将不能访问任何 IPv6 应用。

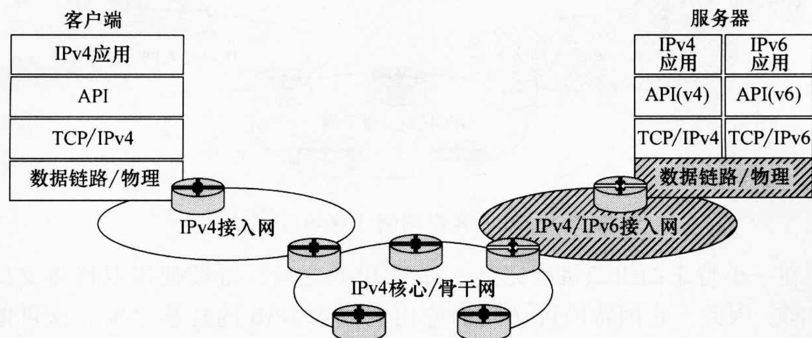


图 15-30　服务器侧迁移场景[147]

（2）这个场景也可能反映了通过一个 IPv4 网络（例如 IPv4 因特网）的一个组织机构间的连接。

1）正在向 IPv6 迁移或已完成迁移的一个组织机构，将可能为面向因特网的 IP 应用（例如它的 web 服务器）实现一台双栈服务器。在这种情形中，一个 IPv4 浏览器客户端可通过因特网访问 web 服务器。web 服务器可同时服务 IPv4 客户端和 IPv6 浏览器客户端。取决于 ISP 的能力，该 ISP 可提供从一个 IPv6 接入网络的转换网关，或可使用打隧道方法。

2）正在向 IPv6 迁移或已完成迁移的一个组织机构，如果它要求到仅运行 IPv4 的合作伙伴的网络连接，那么也可映射到这个场景。实现配置的隧道将是这样一条组织机构间链路（link）的良好方法。

（3）如果我们忽略服务器上双栈的 IPv4 部分，并认为服务器仅支持 IPv6，那么这个场景可代表一台仅支持 IPv4 的客户端尝试访问一台仅支持 IPv6 的服务器。这样一个场景将要求使用如下技术之一：

1）一个 IPv4-IPv6 转换网关处在 IPv4-IPv6 网络边界，这里假定服务器接入网络

也是仅支持 IPv6 的。

2）如果服务器接入网络仅支持 IPv4，那么服务器必须采用一种基于主机的转换机制（例如 BIS 或 BIA），以便在 IPv4 和 IPv6 首部之间转换。

15.6.5 客户端侧迁移

图 15-31 表示以图 15-28 我们的初始配置开始的一个场景，接着将客户端和接入网络路由器都迁移到双栈实现。现有的 IPv4 客户端设备将被增补 IPv6 应用、API 和 TCP/IP 协议栈。当迁移到 Windows XP SP1、Vista 或 7，或 Mac OS X 时，就已经提供了这些功能中的多数增补功能。这个场景代表如下范例情形。

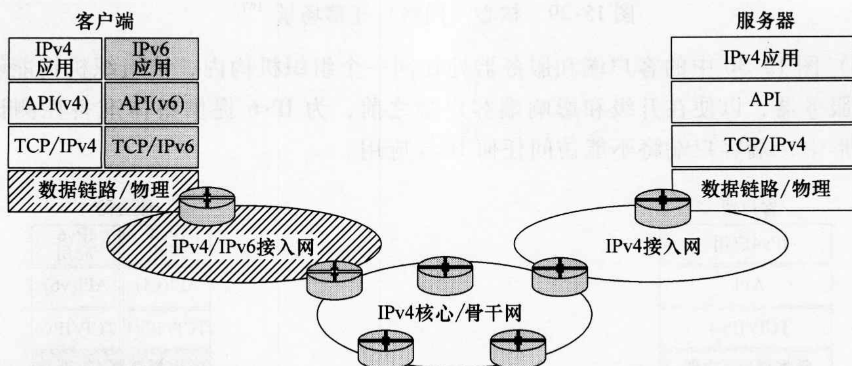


图 15-31 客户端侧迁移场景[147]

（1）在一个特定的组织机构完全迁移到 IPv6 之后，继续使用双栈将支持对 IPv4 网站的访问。因此，在因特网可访问的应用完成到 IPv6 的迁移之前，这可能是实际上会存在多年的后迁移场景。但是，这样一个场景的一种更可能配置，是在组织机构内完全迁移到 IPv6，到 IPv4 因特网（ISP）链路处配置一台转换网关。

（2）一个给定组织机构内的这样一种配置，注定不会落在研究 IPv6 项目的那些组织机构范围之外，他们具有访问外部 IPv6 应用的需要。典型情况下，在一个组织机构内大规模地部署客户端，将要求必要的服务器侧支持。

（3）忽略客户端协议栈的 IPv4 部分，并认为客户端仅支持 IPv6，那么这个场景可勾勒出一名完全迁移的 IPv6 用户访问一项 IPv4 应用（例如一台 web 服务器）的情形。典型情况下，这样一个场景的特征是在 IPv4 ISP 连接上配置一台转换网关，虽然此时也可采用 6to4、ISATAP 或 Teredo 打隧道方法。

15.6.6 客户端-服务器迁移

在一个组织机构内 IPv6 迁移的一种普遍被人期望的方法，其特征是，在大约相同时间或在滚动升级的基础上，对客户端和服务器进行升级，在合适情况下也包括应用的升级。双栈部署方法的使用，便于随时间推移将双栈部署到客户端和服务器。对于应用的迁移，特别是由大量用户群访问的集成企业应用的迁移，必须给予特别的考虑。在迁移中对混合 IPv4/IPv6 客户端的支持会是理想情况，但也许并不现实（见图 15-32）。

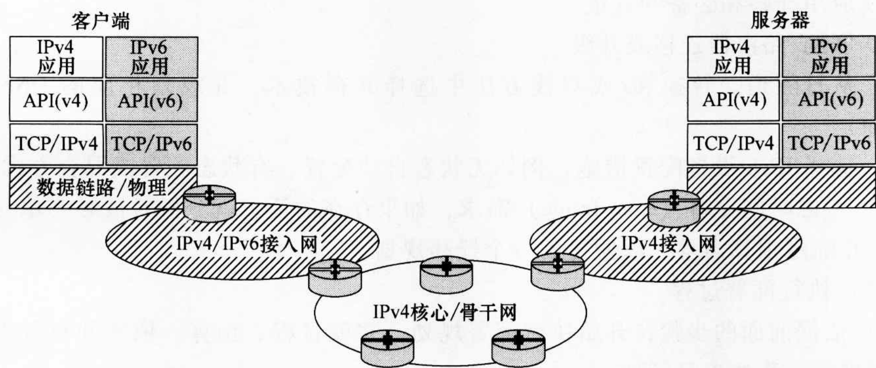


图 15-32　双栈部署场景[147]

这个场景反映了上面描述的企业内范例情况，也反映了依据下面的表中的更广泛范例集合，同时考虑了单栈情形。

15.6.7　总体 IPv6 实现规划

除了地址长度和格式的基本差异外，在其他方面，IPv4 和 IPv6 都是网络层协议，它们使应用能够在共同的层 1 和层 2 网络上进行通信。这样就便于在同一物理网络上迁移过程期间 IPv4 和 IPv6 的共存。但是，层 3 以上的各层将极可能受到影响，特别是显示、利用或支持 IP 地址表项的那些应用更会受到影响。

IPv6 实现提出了许多挑战。这些挑战的关键是为 IPv6 网络分配和部署而定义和组织一个整体规划。这正是一个 IPAM 系统可有助益的领域（discipline）。虽然存在许多涉及的详细步骤，但为规划和实施这样一次迁移，建议采用如下 4 个高层次的步骤。

（1）对您当前的环境建立基线；通过实施我们在本书通篇讨论过的实践，可完成这一步骤。

1）在如下方面对当前 IPv4 网络环境建立清单和做基线：确定哪些 IPv4 地址空间正在使用，它们是如何分配的，哪些个体 IPv4 地址已被指派且正在使用，以及依据设备不同的其他 IP 有关信息。

2）作为前一步骤的必然结果，对相关联的动态主机配置协议（DHCP）服务器配置（指地址池以及相关联策略和选项）建立清单和做基线。

3）识别和记录相关联的域名服务器（DNS）配置以及与 IPv4 设备关联的资源记录。

4）对网络设备（基础设施和端用户）建立清单，以便估计当前和期望的未来 IPv4/IPv6 支持水平。

5）分析应用潜在迁移的影响，如果存在影响的话，那么就对解决这些影响做出规划。

（2）规划您的 IPv6 部署

1）设计（map out）IPv6 实现的一项战略措施，包括如下方面的考虑：

① 应用迁移和必备的升级

② 联网/路由器迁移或升级

③ 从打隧道、转换和/或双栈方法中选择共存技术，并规划相应的 DNS 影响（impact）。

④ 选择 IPv6 设备配置措施，例如无状态自动配置、有状态配置或混合方式

2）考虑时间窗口（time frame）需求，如果存在的话，就分析对被影响单元的依赖性，并确定预算需求，以便得到一个迁移规划。

（3）执行部署过程

1）依据前面的步骤，开始执行部署规划，并就日程、预算、依赖和意外事件计划方面进行过程的项目管理。

2）对做过基线的 IPv4 空间和相关联的 IPv6 重叠（overlay）空间的 IP 地址清单进行跟踪。当然，IPv4 空间中的增长和变化将极可能在迁移过程中持续，所以在合适的情况下，将这些更新嵌入到规划之中。

3）更新 DNS 和 DHCP 服务，以便应对迁移中的环境，例如要支持 IPv4 和 IPv6 主机以及 IPv4 和 IPv6 传输。

（4）管理 IPv4-IPv6 网络

1）管理 IPv4 和 IPv6 地址空间以及相应的 DHCP 和 DNS 服务。

2）依据您的规划，一些 IPv4 空间可能退役。当您网络的一些部分完全迁移到 IPv6 或在网络上作为最终的退役步骤，可能完成这项工作。

3）取决于服务外部主机的需求（依据内部策略，它们可能仅支持 IPv4），您可能期望在您的整个网络上或网络的一些部分上保留 IPv4 协议继续运行。

参考文献

1. J. Postel. *DoD Standard Internet Protocol.* IETF, January 1980. RFC 760.
2. J. Postel. *Internet Protocol.* IETF, September 1981. RFC 791.
3. Internet Systems Consortium (ISC). The ISC Domain Survey. *www.isc.org.* [Online] [Cited: May 15, 2010] www.isc.org/solutions/survey.
4. R. Hinden. *Applicability Statement for the Implementation of Classless Inter-Domain Routting (CIDR).* IETF, September 1993. RFC 1517.
5. Y. Rekhter, T. Li. *An Architecture for IP Address Allocation with CIDR.* IETF, September 1993. RFC 1518.
6. V. Fuller, T. Li, J. Yu, K. Varadhan. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy.* IETF, September 1993. RFC 1519.
7. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. *Address Allocation for Private Internets.* IETF, February 1996. RFC 1918.
8. IANA. *Special-Use IPv4 Addresses.* IETF, September 2002. RFC 3330.
9. M. Cotton, L. Vegoda. *Special Use IPv4 Addresses.* IETF, January, 2010. RFC 5735.
10. S. Deering, R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification.* IETF, December 1998. RFC 2460.
11. T. Rooney. *Introduction to IP Address Management.* IEEE Press/Wiley, 2010.
12. R. Hinden, S. Deering. *IP Version 6 Addressing Architecture.* IETF, February 2006. RFC 4291.
13. Internet Assigned Numbers Authority (IANA). Internet Protocol Version 6 Address Space. *www.iana.org.* [Online] [Cited: May 3, 2010.] http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml.
14. R. Hinden, S. Deering, E. Nordmark. *IPv6 Global Unicast Address Format.* IETF, August 2003. RFC 3587.
15. R. Hinden, B. Haberman. *Unique Local IPv6 Unicast Addresses.* IETF, October 2005. RFC 4193.
16. B. Haberman, D. Thaler. *Unicast-Prefix-Based IPv6 Multicast Addresses.* IETF, August 2002. RFC 3306.
17. J.-S. Park, M.-K. Shin, H.-J. Kim. *A Method for Generating Link-Scoped IPv6 Multicast Addresses.* IETF, April 2006. RFC 4489.

18. M. Crawford, B. Haberman, Eds. *IPv6 Node Information Queries*. IETF, August 2006. RFC 4620.
19. Microsoft. IPv6 Address Autoconfiguration. *www.microsoft.com*. [Online] [Cited: October 19, 2009.] http://msdn.microsoft.com/en-us/library/aa917171.aspx.
20. D. Johnson, S. Deering. *Reserved IPv6 Subnet Anycast Addresses*. IETF, March 1999. RFC 2526.
21. J. Loughney, Ed. *IPv6 Node Requirements*. IETF, April 2006. RFC 4294.
22. M. Blanchet. *A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block*. IETF, April 2003. RFC 3531.
23. IANA. Number Resources. *www.iana.org*. [Online] [Cited: October 20, 2009.] http://www.iana.org/numbers/.
24. K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg, J. Postel. *Internet Registry IP Allocation Guidelines*. IETF, November 1996. RFC 2050.
25. AfriNIC. AfriNIC Home Page. *www.afrinic.net*. [Online] [Cited: October 20, 2009.] http://www.afrinic.net/.
26. APNIC. APNIC Home Page. *www.apnic.net*. [Online] [Cited: October 20, 2009.] http://www.apnic.net/.
27. ARIN. ARIN Home Page. *www.arin.net*. [Online] [Cited: October 20, 2009.] http://www.arin.net/.
28. LACNIC. LACNIC Home Page. *www.lacnic.net*. [Online] http://www.lacnic.net/.
29. RIPE NCC. RIPE Network Coordination Centre Home Page. *www.ripe.net*. [Online] [Cited: October 20, 2009.] http://www.ripe.net/.
30. C. Huitema. *The H Ratio for Address Assignment Efficiency*. IETF, November 1994. RFC 1715.
31. A. Durand, C. Huitema. *The Host-Density Ratio for Address Assignment Efficiency: An Update on the H Ratio*. IETF, November 2001. RFC 3194.
32. R. Droms. *Dynamic Host Configuration Protocol*. IETF, March 1997. RFC 2131.
33. S. Alexander, R. Droms. *DHCP Options and BOOTP Vendor Extensions*. IETF, March 1997. RFC 2132.
34. B. Croft, J. Gilmore. *Bootstrap Protocol (BOOTP)*. IETF, September 1985. RFC 951.
35. Internet Systems Consortium. *dhcpd.conf man*. Redwood City, CA: Internet Systems Consortium, Inc. (ISC), 2010.
36. T. Lemon, S. Cheshire, B. Volz. *The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) Version 4*. IETF, December 2002. RFC 3442.
37. R. Droms, K. Fong. *NetWare/IP Domain Name and Information*. IETF, November 1997. RFC 2242.
38. G. Stump, R. Droms, Y. Gu, R. Vyaghrapuri, A. Demirtjis, B. Beser, J. Privat. *The User Class Option for DHCP*. IETF, November 2000. RFC 3004.
39. C. Perkins, E. Guttman. *DHCP Options for Service Location Protocol*. IETF, June 1999. RFC 2610.
40. S. Park, P. Kim, B. Volz. *Rapid Commit Option for the Dynamic Host Configuration Protocol Version 4 (DHCPv4)*. IETF, March 2005. RFC 4039.
41. M. Stapp, B. Volz, Y. Rekhter. *The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option*. IETF, October 2006. RFC 4702.
42. M. Patrick. *DHCP Relay Agent Information Option*. IETF, January 2001. RFC 3046.

43. C. Monia, J. Tseng, K. Gibbons. *The IPv4 Dynamic Host Configuration Protocol (DHCP) Option for the Internet Storage Name Service*. IETF, September 2005. RFC 4174.
44. R. Droms. *Unused Dynamic Host Configuration Protocol (DHCP) Option Codes*. IETF, January 2004. RFC 3679.
45. D. Provan. *DHCP Options for Novell Directory Services*. IETF, November 1997. RFC 2241.
46. K. Chowdhury, P. Yegani, L. Madour. *Dynamic Host Configuration Protocol (DHCP) Options for Broadcast and Multicast Control Servers*. IETF, November 2005. RFC 4280.
47. R. Droms, W. Arbaugh, Eds. *Authentication for DHCP Messages*. IETF, June 2001. RFC 3118.
48. R. Woundy, K. Kinnear. *Dynamic Host Configuration Protocol (DHCP) Leasequery*. IETF, February 2006. RFC 4388.
49. M. Johnston, S. Venaas, Eds. *Dynamic Host Configuration Protocol (DHCP) Options for Intel Preboot eXecution Environment (PXE)*. IETF, November 2006. RFC 4578.
50. S. Drach. *DHCP Option for The Open Group's User Authentication Protocol*. IETF, January 1999. RFC 2485.
51. H. Schulzrinne. *Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information*. IETF, November 2006. RFC 4776.
52. E. Lear, P. Eggert. *Timezone Options for DHCP*. IETF, April 2007. RFC 4833.
53. R. Troll. *DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients*. IETF, May 1999. RFC 2563.
54. C. Smith. *The Name Service Search Option for DHCP*. IETF, September 2000. RFC 2937.
55. G. Waters. *The IPv4 Subnet Selection Optoin for DHCP*. IETF, November 2000. RFC 3011.
56. B. Aboba, S. Cheshire. *Dynamic Host Configuration Protocol (DHCP) Domain Search Option*. IETF, November 2002. RFC 3397.
57. H. Schulzrinne. *Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers*. IETF, August 2002. RFC 3361.
58. P. Agarwal, B. Akyol. *The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) Version 4*. IETF, December 2002. RFC 3442.
59. B. Beser, P. Duffy, Eds. *Dynamic Host Configuration Protocol (DHCP) Option for CableLabs Client Configuration*. IETF, March 2003. RFC 3495.
60. J. Polk, J. Schnizlein, M. Linsner. *Dynamic Host Configuration Protocol Option for Coordinate-Based Location Configuration Information*. IETF, July 2004. RFC 3825.
61. J. Littlefield. *Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol Version 4 (DHCPv4)*. IETF, October 2004. RFC 3925.
62. L. Morand, A. Yegin, S. Kumar, S. Madanapalli. *DHCP Options for Protocol for Carrying Authentication for Network Access (PANA) Authentication Agents*. IETF, May 2008. RFC 5192.
63. H. Schulzrinne, J. Polk, H. Tschofenig. *Discovering Location-to-Service Translation (LoST) Servers Using the Dynamic Host Configuration Protocol (DHCP)*. IETF, August 2008. RFC 5223.
64. P. Calhoun. *Control and Provisioning of Wireless Access Points (CAPWAP) Access Controller DHCP Option*. IETF, March 2009. RFC 5417.
65. G. Bajko, S. Das. *Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Options for IEEE 802.21 Mobility Services (MoS) Discovery*. IETF, December 2009. RFC 5678.

66. B. Volz. *Reclassifying Dynamic Host Configuration Protocol Version 4 (DHCPv4) Options*. IETF, November 2004. RFC 3942.

67. D. Hankins. *Dynamic Host Configuration Protocol Options Used by PXELINUX*. IETF, December 2007. RFC 5071.

68. R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. IETF, July 2003. RFC 3315.

69. H. Schulzrinne, B. Volz. *Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers*. IETF, July 2003. RFC 3319.

70. R. Droms, Ed. *DNS Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. IETF, December 2003. RFC 3646.

71. O. Troan, R. Droms. *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) Version 6*. IETF, December 2003. RFC3633.

72. V. Kalusivalingam. *Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. IETF, October 2004. RFC 3898.

73. V. Kalusivalingam. *Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6*. IETF, May 2005. RFC 4075.

74. S. Venaas, T. Chown, B. Volz. *Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*. IETF, November 2005. RFC 4242.

75. B. Volz. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option*. IETF, August 2006. RFC 4649.

76. B. Volz. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Subscriber-ID Option*. IETF, June 2006. RFC 4580.

77. B. Volz. *The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option*. IETF, October 2006. RFC 4704.

78. S. Zeng, B. Volz, K. Kinnear, J. Brzozowski. *DHCPv6 Relay Agent Echo Request Option*. IETF, September 2007. RFC 4994.

79. J. Brzozowski, K. Kinnear, B. Volz, S. Zeng. *DHCPv6 Leasequery*. IETF, September 2007. RFC 5007.

80. H. Jang, A. Yegin, K. Chowdhury, J. Choi. *DHCP Options for Home Information Discovery in MIPv6*. IETF, May 2008. draft-ietf-mip6-hiopt-17.

81. M. Stapp *DHCPv6 Bulk Leasequery*. IETF, February 2009. RFC 5460.

82. D. Jones, R. Woundy. *The DOCSIS (Data-Over-Cable Service Interface Specifications) Device Class DHCP (Dynamic Host Configuration Protocol) Relay Agent Information Sub-Option*. IETF, April 2002. RFC 3256.

83. K. Kinnear, M. Stapp, R. Johnson, J. Kumarasamy. *Link Selection Sub-Option for the Relay Agent Information Option for DHCPv4*. IETF, April 2003. RFC 3527.

84. R. Johnson, T. Palaniappan, M. Stapp. *Subscriber-ID Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option*. IETF, March 2005. RFC 3993.

85. R. Droms, J. Schnizlein. *Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP)*. IETF, February 2005. RFC 4014.

86. M. Stapp, T. Lemon. *The Authentication Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option*. IETF, March 2005. RFC 4030.

87. M. Stapp, R. Johnson, T. Palaniappan. *Vendor-Specific Information Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option*. IETF, December 2005. RFC 4243.

88. K. Kinnear, M. Normoyle, M. Stapp. *The Dynamic Host Configuration Protocol Version 4 (DHCPv4) Relay Agent Flags Suboption*. IETF, September 2007. RFC 5010.

89. J. Jumarasamy, K. Kinnear, M. Stapp. *DHCP Server Identifier Override Suboption*. IETF, February 2008. RFC 5107.

90. P. Valian. NetReg. *sourceforge.net*. [Online] [Cited: January 31, 2010.] http://netreg.sourceforge.net.

91. Cisco Systems, Inc., *Cisco Network Admission Control (NAC)*. San Jose, CA: Cisco Systems, Inc., 2009.

92. Microsoft Corporation. *Introduction to Nework Access Protection*. Redmond, WA: Microsoft Corporation, 2008.

93. H. Eidnes, G. de Groot, P. Vixie. *Classless IN-ADDR.ARPA Delegation*. IETF, March 1998. RFC 2317.

94. P. Faltstrom, P. Hoffman, A. Costello. *Internationalizing Domain Names in Applications (IDNA)*. IETF, March 2003. RFC 3490.

95. P. Hoffman, M. Blanchet. *Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)*. IETF, March 2003. RFC 3491.

96. P. Hoffman, M. Blanchet. *Preparation of Internationalized Strings (stringprep)*. IETF, December 2002. RFC 3454.

97. A. Costello. *Punycode: A Bootstring Encoding of Unicode for Internationalized Domain Names in Applications (IDNA)*. IETF, March 2003. RFC 3492.

98. Internationalized Domain Names (IDN). *International Telecommunications Union*. [Online] [Cited: October 21, 2009.] http://www.itu.int/ITU-T/special-projects/idn/introduction.html.

99. P. V. Mockapetris. *Domain Names—Implementation and Specification*. IETF, November 1987. RFC 1035.

100. P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE)*. IETF, April 1997. RFC 2136.

101. P. Vixie. *Extension Mechanisms for DNS (EDNS0)*. IETF, August 1999. RFC 2671.

102. P. Vixie, O. Gudmundsson, D. Eastlake, 3rd, B. Wellington. *Secret Key Transaction Authentication for DNS (TSIG)*. IETF, May 2000. RFC 2845.

103. D. Eastlake, 3rd. *Secret Key Establishment for DNS (TKEY RR)*. IETF, September 2000. RFC 2930.

104. D. Eastlake, 3rd. *HMAC SHA TSIG Algorithm Identifiers*. IETF, August 2006. RFC 4635.

105. D. Eastlake, 3rd. *Domain Name System (DNS) IANA Considerations*. IETF, November 2008. RFC 5395.

106. Internet Assigned Numbers Authority, (IANA). Domain Name System (DNS) Parameters. *www.iana.org*. [Online] [Cited: August 10, 2010.] http://www.iana.org/assignments/dns-parameters.

107. M. Crawford. *Non-Terminal DNS Name Redirection*. IETF, August 1999. RFC 2672.

108. C. F. Everhart, L. A. Mamakos, R. Ullmann, P. V. Mockapetris. *New DNS RR Definitions*. IETF, October 1990. RFC 1183.

109. M. Mealling. *Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database*. IETF, October 2002. RFC 3403.

110. M. Mealling. *Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm*. IETF, October 2002. RFC 3402.

111. P. Faltstrom, M. Mealling. *The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)*. IETF, April 2004. RFC 3761.

112. M. Wong, W. Schlitt. *Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1*. IETF, April 2006. RFC 4408.

113. E. Allman, J. Fenton, M. Delany, J. Levine. *DomainKeys Identified Mail (DKIM) Author Signing Practices (ADSP)*. IETF, August 2009. RFC 5617.

114. R. Arends, R. Austein, M. Larson, D. Massey, S. Rose. *Resource Records for DNS Security Extensions*. IETF, March 2005. RFC 4034.

115. M. Andrews, S. Weiler. *The DNSSEC Lookaside Validation (DLV) DNS Resource Record*. IETF, February 2006. RFC 4431.

116. S. Josefsson. *Storing Certificates in the Domain Name System (DNS)*. IETF, March 2006. RFC 4398.

117. M. Richardson. *A Method for Storing IPsec Keying Material in DNS*. IETF, March 2005. RFC 4025.

118. D. Eastlake, 3rd. *DNS Request and Transaction Signatures (SIG(0)s)*. IETF, September 2000. RFC 2931.

119. C. Farrell, M. Schulze, S. Pleitner, D. Baldoni. *DNS Encoding of Geographical Location*. IETF, November 1994. RFC 1712.

120. C. Davis, P. Vixie, T. Goodwin, I. Dickinson. *A Means for Expressing Location Information in the Domain Name System*. IETF, January 1996. RFC 1876.

121. C. Allocchio, *Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)*. IETF, January 1998. RFC 2163.

122. M. Crawford, C. Huitema. *DNS Extensions to Support IPv6 Address Aggregation and Renumbering*. IETF, July 2000. RFC 2874.

123. S. Thomson, C. Huitema, V. Ksinant, M. Souissi. *DNS Extensions to Support IP Version 6*. IETF, October 2003. RFC 3596.

124. P. Koch. *A DNS RR Type for Lists of Address Prefixes*. IETF, June 2001. RFC 3123.

125. ATM Forum Technical Committee. *ATM Name System, V2.0*. ATM Forum, 2000. AF-DANS-0152.000.

126. M. Stapp, T. Lemon, A. Gustafsson. *A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR)*. IETF, October 2006. RFC 4701.

127. P. Nikander, J. Laganier. *Host Identity Protocol (HIP) Domain Name System (DNS) Extensions*. IETF, April 2008. RFC 5205.

128. D. Eastlake, 3rd. *DSA KEYs and SIGs in the Domain Name System (DNS)*. IETF, March 1999. RFC 2536.

129. R. Atkinson. *Key Exchange Delegation Record for the DNS*. IETF, November 1997. RFC 2230.

130. B. Manning, R. Colella. *DNS NSAP Resource Records*. IETF, October 1994. RFC 1706.

131. B. Laurie, G. Sisson, R. Arends, D. Blacka. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*. IETF, March 2008. RFC 5155.

132. S. Weiler. *Legacy Resolver Compatibility for Delegation Signer (DS)*. IETF, May 2004. RFC 3755.

133. R. Gellens, J. Klensin. *Message Submission for Mail*. IETF, April 2006. RFC 4409.

134. A. Gulbrandsen, P. Vixie, L. Esibov. *A DNS RR for Specifying the Location of Services (DNS SRV)*. IETF, February 2000. RFC 2782.

135. J. Schlyter, W. Griffin. *Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*. IETF, January 2006. RFC 4255.

136. Internet Corporation for Assigned Names and Numbers (ICANN). Factsheet—Root server attack on 6 February 2007. *Internet Corporation for Assigned Names and Numbers. www.icann.org*. [Online] [Cited: October 22, 2009.] http://www.icann.org/announcements/factsheet-dns-attack-08mar07.pdf.

137. B. Woodcock.Best Practices in IPv4 Anycast Routing. *Packet Clearing House*. [Online] August 2002. [Cited: May 3, 2010.] http://www.pch.net/resources/papers/ipv4-anycast/ipv4-anycast.pdf.

138. T. Rooney. *DNS Anycast Addressing for High Availability and Performance*. Santa Clara, CA: BT INS, Inc., 2008.

139. D. Atkins, R. Austein. *Threat Analysis of the Domain Name System (DNS)*. IETF, August 2004. RFC 3833.

140. D. Eastlake, 3rd. *Domain Name System Security Extensions*. IETF, March 1999. RFC 2535.

141. R. Arends, R. Austein, M. Larson, D. Massey, S. Rose. *DNS Security Introduction and Requirements*. IETF, March 2005. RFC 4033.

142. R. Arends, R. Austein, M. Larson, D. Massey, S. Rose. *Protocol Modifications for the DNS Security Extensions*. IETF, March 2005. RFC 4035.

143. M. StJohns. *Automated Updates of DNS Security (DNSSEC) Trust Anchors*. IETF, September 2007. RFC 5011.

144. Internet Systems, Consortium. *BIND 9 Administrator Reference Manual*. Redwood City, CA: Internet Systems Consortium, Inc. (ISC), 2010.

145. A. Cherenson. *nslookup man page (BIND Distribution)*. Redwood City, CA: Internet Systems Consortium, Inc. (ISC), 2010.

146. Internet Systems, Consortium. *dig man page (with BIND Distribution)*. Redwood City, CA: Internet Systems Consortium (ISC), 2010.

147. T. Rooney. *IPv4-to-IPv6 Transition and Co-Existence Strategies*. Santa Clara, CA: BT INS, Inc., March 2008.

148. A. Durand, J. Ihren. *DNS IPv6 Transport Operational Guidelines*. IETF, September 2004. RFC 3901.

149. T. Chown, S. Venaas, C. Strauf. *Dynamic Host Configuration Protocol (DHCP): IPv4 and IPv6 Dual-Stack Issues*. IETF, May 2006. RFC 4477.

150. M. Blanchet, F. Parent. *IPv6 Tunnel Broker with Tunnel Setup Protocol (TSP)*. IETF, February 2010. RFC 5572.

151. C. Huitema. *Teredo: Tunneling IPv6 Over UDP Through Network Address Translations (NATs)*. IETF, February 2006. RFC 4380.

152. J. Bound, L. Toutain, J. L. Richier. *Dual Stack IPv6 Dominant Transition Mechanism (DSTM)*. IETF, October 2005. draft-bound-dstm-exp-04.txt.
153. E. Nordmark. *Stateless IP/ICMP Translation Algorithm (SIIT)*. IETF, February 2000. RFC 2765.
154. K. Tsuchiya, H. Higuchi, Y. Atarashi. *Dual Stack Hosts Using the "Bump-in-the-Stack" Technique (BIS)*. IETF, February 2000. RFC 2767.
155. S. Lee, M.-K. Shin, Y.-J. Kim, E. Nordmark, A. Durand. *Dual Stack Hosts Using "Bump-in-the-API" (BIA)*. IETF, October 2002. RFC 3338.
156. G. Tsirtsis, P. Srisuresh. *Network Address Translation-Protocol Translation (NAT-PT)*. IETF, February 2000. RFC 2766.
157. C. Aoun, E. Davies. *Reasons to Move the Network Address Translator-Protocol Translator (NAT-PT) to Historic Status*. IETF, July 2007. RFC 4966.
158. M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones. *SOCKS Protocol Version 5*. IETF, March 1996. RFC 1928.
159. H. Kitamura. *A SOCKS-Based IPv6/IPv4 Gateway Mechanism*. IETF, April 2001. RFC 3089.
160. J. Hagino, K. Yamamoto. *An IPv6-to-IPv4 Transport Relay Translator*. IETF, June 2001. RFC 3142.
161. R. Despres. *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)*. IETF, January 2010. RFC 5569.
162. A. Durand, Ed. *Dual-Stack Lite Broadband Deployments Post IPv4 Exhaustion*. IETF, February 2010. draft-ietf-software-dual-stack-lite-03.txt.
163. J. Klensin, Ed. *Simple Mail Transfer Protocol*. IETF, April 2001. RFC 2821.
164. P. Resnick, Ed. *Internet Message Format*. IETF, April 2001. RFC 2822.
165. A. Drescher. Director, Technical Services. *Private Correspondence*. Throughout 2006–present.
166. T. Rooney. *IPv6 Addressing and Management Challenges*. Santa Clara, CA: BT INS, Inc., March 2008.
167. O. Kolkman, R. Gieben. *DNSSEC Operational Practices*. IETF, September 2006. RFC 4641.
168. Office of Government and Commerce (OGC). ITIL(R) Home. *Official ITIL(R) Website*. [Online] APM Group Limited. [Cited: September 16, 2009.] http://www.itil-officialsite.com/home/home.asp.
169. L. Delgrossi, L. Berger, Eds. *Internet Stream Protocol Version 2 (ST2) Protocol Specification – Version ST2+*. IETF, August 1995. RFC 1819.
170. T. Bates, Y. Rekhter. *Scalable Support for Multi-Home Multi-Provider Connectivity*. IETF, January 1998. RFC 2260.
171. J. Abley, K. Lindqvist, E. Davies, B. Black, V. Gill. *IPv4 Multihoming Practices and Limitations*. IETF, July 2005. RFC 4116.
172. G. Huston. *Architectural Approaches to Multi-Homing for IPv6*. IETF, September 2005. RFC 4177.
173. E. Nordmark, T. Li. *Threats Relating to IPv6 Multihoming Solutions*. IETF, October 2005. RFC 4218.
174. T. Rooney. *Applying ITIL Best Practice Principles to IPAM*. Santa Clara, CA: BT Diamond IP, August 2008.
175. R. Johnson. *TFTP Server Address Option for DHCPv4*. IETF, June 2010. RFC 5859.

176. W. Townsley, O. Troan. *IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) – Protocol Specification*. IETF, August 2010. RFC 5969.

177. K. Nichols, S. Blake, F. Baker, D. Black. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. IETF, December 1998. RFC 2474.

178. M. Thomson, J. Winterbottom. *Discovering the Local Location Information Server (LIS)*. IETF, March 2010. draft-ietf-geopriv-lis-discovery-15.txt.

179. S. Lawrence, Ed., J. Elwell. *Session Initiation Protocol (SIP) User Agent Configuration*. IETF, June 2010. draft-lawrence-sipforum-user-agent-config-03.

180. R. Gayraud, B. Lourdelet. *Network Time Protocol (NTP) Server Option for DHCPv6*. IETF, June 2010. RFC 5908.

181. T. Huth, J. Freimann, V. Zimmer, D. Thaler. *DHCPv6 Options for Network Boot*. IETF, July 2010. draft-ietf-dhc-dhcpv6-opt-netboot-10.txt.

182. M. Crawford. *Binary Labels in the Domain Name System*. IETF, August 1999. RFC 2673.

183. R. Bush, A. Durand, B. Fink, O. Gudmundsson, T. Hain, Editors. *Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS). IETF, August 2002. RFC 3363.*

184. J. Klensin. *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*. IETF, August 2010. RFC 5890.

185. J. Klensin. *Internationalized Domain Names for Applications (IDNA): Protocol*. IETF, August 2010. RFC 5891.

186. P. Faltstrom, Ed. *The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)*. IETF, August 2010. RFC 5892.

187. H. Alvestrand, Ed. Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA). IETF, August 2010. RFC 5893.

188. J. Klensin. *Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale*. IETF, August 2010. RFC 5894.

189. R. Allbery. *DNS SRV Resource Records for AFS*. IETF, April 2010. RFC 5864.

190. J. Levine. *DNS Blacklists and Whitelists*. IETF, February 2010. RFC 5782.

191. R. Austein. *DNS Name Server Identifier (NSID) Option*. IETF, August 2007. RFC 5001.

192. ITU-T Study Group 4. *Series M: TMN and Network Maintenance: International Transmission Systems, Telephone Circuits, Telegraphy, Facsimile and Leased Circuits. TMN Management Functions*. ITU, 2001. ITU-T M.3400.

193. T. Chown. *Use of VLANs for IPv4-IPv6 Coexistence in Enterprise Networks*. IETF, June 2006. RFC 4554.

194. C. Huitema. *An Anycast Prefix for 6to4 Relay Routers*. IETF, June 2001. RFC 3068.

词 汇 表

本书中通篇所用的主要术语汇总如下：

• DHCP（动态主机配置协议）：使 IP 地址到网络主机或设备的指派自动化。DHCP 技术适用于 IPv4 和 IPv6 地址指派，虽然典型情况下，“DHCP”指 IPv4 地址指派。

• DHCPv6：专门用于 IPv6 地址的 DHCP，并不是 DHCP 协议的版本 6。

• DNS（域名系统）：因特网名字、地址和其他信息的分布式数据库。

• DNSSEC：DNS 安全扩展，提供解析数据源发认证（源真正地发布了这个数据）、数据完整性验证（沿路数据没有被修改）和经过认证的数据存在性拒绝（被请求的数据在这个区域中不存在）。

• FCAPS：网络管理的五个主要功能域的首字母缩写，即在 ITU 电信管理网络标准中所描述的故障、配置、记账、性能和安全。

• Host（主机）：在一个 IP 网络上通信的一台端设备，例如一台服务器、笔记本计算机和 VoIP 电话。我们称一台端设备，是相对网络基础设施设备而言的，例如路由器和交换机。

• IP（因特网协议）：在因特网间和所有 IP 网络间使用的网络层。IP 通常指所有 IP 版本，而 IPv4 和 IPv6 指相应的 IP 版本。

• IPAM（IP 地址管理）：管理 IP 地址空间以及相关联 DHCP 和 DNS 服务的系统性（disciplined）方法。

• ISC（因特网系统团体（Consortium））：DHCP 和 DNS（BIND）参考实现的开发者。

• ITIL®（信息技术基础设施库）：由英国商务部（OGC）开发，是一个最佳实践框架，采取的是这样的一个视角，即信息技术（IT）组织机构是企业的一个服务提供商。

• KSK（密钥签名密钥）：在 DNSSEC 内部使用，用来对一个区域签名密钥（ZSK）进行签名，ZSK 接下来对 DNS 区域数据进行签名。公开 KSK 是在相应安全区域内的一条 DNSKEY 资源记录中发布的。被配置信任这个区域之数据的解析器或递归服务器，在其相应配置内必须有公开 KSK 的一个拷贝（或一个父区域或旁查核验器区域的公开 KSK）被配置为信任锚点。

• NAT（网络地址翻译转换）：一台网关或防火墙，在转发报文之前，改变（转换）IP 报文内的一个 IP 地址；常用于企业网络之中，将内部私有 IP 地址转换为外部公开 IP 地址。

• TCP（传输控制协议）：TCP/IP 协议族内面向连接的传输层协议。

• UDP（用户数据报协议）：TCP/IP 协议族内无连接的传输层协议。

• ZSK（区域签名密钥）：用在 DNSSEC 内，对区域信息（指区域内资源记录集内的每条记录）进行签名。

RFC 索 引

本索引列出了因特网工程任务组（IETF）发布的IPAM-有关的主要请求评述（RFC）文档。RFC文档可从www. ietf. org/rfc检索得到。没有列出废弃（Obsoleted）的各RFC。RFC 3789~3796是部署IPv4地址的综述，在下面的表中没有列出，但它们就特定应用的IPv4地址进行的讨论和定义，提供了有助益的深邃理解。

状态栏指明RFC状态，为信息型的、试验型的、标准跟踪、草案标准、建议标准、标准及历史型的。已被采纳为最佳当前实践（BCP）的各RFC以BCP号枚举列出。

IPv4协议各RFC

RFC	状态	标　　题
791	标准	因特网协议
1042	标准	在IEEE 802网络上传输IP数据报的标准
1546	信息型的	主机任意播服务
1878	历史型的	IPv4的变长子网表
2101	信息型的	如今的IPv4地址行为
2365	BCP 23	管理范围的IP组播
3927	建议标准	IPv4链路本地地址的动态配置
4116	信息型的	IPv4多穴连接实践和限制
4632	BCP 122	无类域间路由(CIDR):因特网地址指派和汇聚规划

IPv6协议各RFC

RFC	状态	标　　题
1752	建议标准	IP下一代协议的建议
1881	信息型的	IPv6地址分配管理
1887	信息型的	IPv6单播地址分配架构
2375	信息型的	IPv6组播地址指派
2460	草案标准	因特网协议版本6(IPv6)规范
2526	建议标准	预留的IPv6子网任意播地址
3484	建议标准	因特网协议版本6(IPv6)的默认地址选择
3582	信息型的	IPv6站点多穴连接架构的目标
3587	信息型的	IPv6全球单播地址格式
3627	信息型的	在路由器之间使用/127前缀长度被认为是有害的
3701	信息型的	6bone(IPv6测试地址分配)分阶段停用

（续）

RFC	状态	标　　题
3879	建议标准	废弃的站点本地地址
3956	建议标准	在一个 IPv6 组播地址中内嵌会聚点(RP)地址
4007	建议标准	IPv6 有范围约束的地址架构
4076	信息型的	IPv6 无状态动态主机配置协议的重新编址需求(DHCPv6)
4177	信息型的	IPv6 多穴连接的架构性方法
4193	建议标准	唯一本地 IPv6 单播地址
4218	信息型的	与 IPv6 多穴连接解决方案有关的威胁
4291	草案标准	IP 版本 6 寻址架构
4294	信息型的	IPv6 节点需求
4339	信息型的	IPv6 主机的 DNS 服务器信息配置方法
4489	建议标准	生成链路范围 IPv6 组播地址的一种方法
4843	试验型的	覆盖可路由的密码学哈希标识符(ORCHID)的 IPv6 前缀
4861	草案标准	IP 版本 6(IPv6)的邻居发现
4862	草案标准	IPv6 无状态地址自动配置
4941	草案标准	IPv6 中无状态地址自动配置的隐私扩展
4968	信息型的	基于 802.16 网络的 IPv6 链路模型分析
5006	试验型的	DNS 配置的 IPv6 路由器通告选项
5156	信息型的	特殊用途的 IPv6 地址
5157	信息型的	网络扫描的 IPv6 隐含意义
5375	信息型的	IPv6 单播地址指派考虑因素
5453	标准跟踪	预留的 IPv6 接口标识符
5902	信息型的	有关 IPv6 网络地址转换的 IAB 思考

IPv4/IPv6 共存各 RFC

RFC	状态	标　　题
2185	信息型的	IPv6 迁移的路由特征(Aspects)
2529	建议标准	没有显式隧道条件下 IPv4 域之上的 IPv6 传输
2765	建议标准	无状态 IP/ICMP 转换算法(SIIT)
2767	信息型的	使用“协议栈肿块”(BIS)技术的双栈主机
3053	信息型的	IPv6 隧道代理
3056	建议标准	通过 IPv4 网络云的 IPv6 域间连接[6to4]
3068	建议标准	6to4 中继路由器的一个任意播前缀
3089	信息型的	一种基于 SOCKS 的 IPv6/IPv4 网关机制
3142	信息型的	一种 IPv6 到 IPv4 传输中间转换器

（续）

RFC	状态	标　　题
3338	试验型的	使用“API 肿块”(BIA)的双栈主机
3574	信息型的	3GPP 网络的迁移场景
3750	信息型的	无管理(Unmanaged)网络 IPv6 迁移场景
3904	信息型的	无管理网络的 IPv6 迁移机制评估
3964	信息型的	6to4 安全考虑
3974	信息型的	混合 IPv4/IPv6 环境中 SMTP 运行经验
4029	信息型的	将 IPv4 引入 ISP 网络的场景和分析
4038	信息型的	IPv6 迁移的应用特征
4057	信息型的	IPv6 企业网络场景
4213	建议标准	IPv6 主机和路由器的基本迁移机制
4215	信息型的	第三代合作伙伴项目(3GPP)网络中 IPv6 迁移的分析
4241	信息型的	IPv6/IPv4 双栈因特网接入服务的一种模型
4361	建议标准	动态主机配置协议版本 4(DHCPv4)的节点特定客户端标识符
4380	建议标准	Teredo:通过网络地址转换(NAT)的 UDP 上 IPv6 隧道传输
4477	信息型的	动态主机配置协议(DHCP):IPv4 和 IPv6 双栈问题
4554	信息型的	在企业网中 IPv4 - IPv6 共存而使用 VLAN
4798	建议标准	使用 IPv6 提供商边缘路由器(6PE)在 IPv4 MPLS 上连接 IPv6 孤岛
4852	信息型的	IPv6 企业网分析——IP 层 3 聚焦
4942	信息型的	IPv6 迁移/共存的安全考虑
4966	信息型的	将网络地址转换器——协议转换器(NAT - PT)移到历史型状态的原因
4977	信息型的	问题陈述:双栈移动性
5181	信息型的	在 802.16 网络中的 IPv6 部署场景
5211	信息型的	一项因特网迁移计划
5214	信息型的	站点内自动隧道寻址协议(ISATAP)
5569	信息型的	在 IPv4 基础设施上的 IPv6 快速部署(6rd)
5747	试验型的	使用 IP 封装和 MP - BGP 扩展的 4over6 迁移(Transit)解决方案

IP 地址分配的各 RFC

RFC	状态	标　　题
1219	信息型的	子网号码指派
1518	历史型的	采用 CIDR 的 IP 地址分配架构
1900	信息型的	重新编址需要进行研究
1715	信息型的	地址指派效率的 H 比率
1918	BCP 5	私有因特网的地址分配

（续）

RFC	状态	标　　题
2008	BCP 7	因特网路由各种地址分配策略的隐含意义
2050	BCP 12	因特网注册处 IP 分配指南
2071	信息型的	网络重新编号综述：我为什么想要它及它到底是什么？
2908	信息型的	因特网组播地址分配架构
3177	信息型的	将 IPv6 地址分配到站点的 IAB/IESG 建议
3194	信息型的	地址指派效率的 H 密度比率——H 比率的更新
3531	信息型的	管理一个 IPv6 地址块的各比特指派的一种灵活方法
3819	BCP 89	对因特网子网设计人员的忠告
3849	信息型的	为归档（Documentation）预留的 IPv6 地址前缀
4147	信息型的	针对 IANA IPv6 注册处建议的格式变更
4192	信息型的	在没有纪念日（指胜利）条件下对一个 IPv6 网络重新编号的规程
4779	信息型的	在宽带接入网络中的 ISP IPv6 部署场景
4786	BCP 126	任意播服务的运行
5505	信息型的	因特网主机配置的原则
5684	信息型的	采用重叠地址空间部署 NAT 的意外后果
5735	BCP 153	特殊用途的 IPv4 地址
5736	信息型的	IANA IPv4 特殊目的地址注册处
5737	信息型的	为归档（Documentation）预留的 IPv4 地址块
5771	BCP 51	IPv4 组播地址指派 IANA 指南
58871	信息型的	重新编址仍然需要进行研究工作

DHCP 协议各 RFC

RFC	状态	标　　题
1534	草案标准	DHCP 和 BOOTP 之间的互操作
2131	草案标准	动态主机配置协议
2132	草案标准	DHCP 选项和 BOOTP 厂商扩展
2241	建议标准	Novell 目录服务的 DHCP 选项
2242	建议标准	NetWare/IP 域名和信息
2485	建议标准	开放群组用户认证协议的 DHCP 选项
2563	建议标准	在 IPv4 客户端禁止无状态自动配置的 DHCP 选项
2610	建议标准	服务定位协议的 DHCP 选项
2855	建议标准	用于 IEEE 1394 的 DHCP
2937	建议标准	DHCP 的名字服务搜索选项
3004	建议标准	DHCP 的用户类选项

（续）

RFC	状态	标　　题
3011	建议标准	DHCP 的 IPv4 子网选择选项
3046	建议标准	DHCP 中继代理信息选项
3074	建议标准	DHC 负载均衡算法
3118	建议标准	DHCP 消息认证
3203	建议标准	DHCP 重新配置扩展
3256	建议标准	DOCSIS 设备类 DHCP 中继代理信息子选项
3361	建议标准	会话初始协议(SIP)服务器的动态主机配置协议(用于 IPv4 的 DHCP)选项
3396	建议标准	动态主机配置协议(DHCPv4)中的编码长(long)选项
3397	建议标准	动态主机配置协议(DHCP)域搜索选项
3442	建议标准	动态主机配置协议(DHCP)版本 4 的无类静态路由选项
3456	建议标准	动态主机配置(DHCPv4)的 IPsec 隧道模式配置
3495	建议标准	CableLabs 客户端配置的动态主机配置协议(DHCP)选项
3527	建议标准	用于 DHCPv4 的中继代理信息选项的链路选择子选项
3634	建议标准	用于动态主机配置协议(DHCP)CableLabs 客户端配置(CCC)选项的密钥分发中心(KDC)服务器地址子选项
3679	信息型的	未用动态主机配置协议(DHCP)选项代码
3825	建议标准	基于协作的位置配置信息的动态主机配置协议选项
3925	建议标准	用于动态主机配置协议版本 4(DHCPv4)的厂商识别的厂商选项
3942	建议标准	重新分类动态主机配置协议版本 4(DHCPv4)选项
3993	建议标准	用于动态主机配置协议(DHCP)中继代理选项的订户 – ID 子选项
4030	建议标准	用于动态主机配置协议(DHCP)中继代理选项的认证子选项
4039	建议标准	用于动态主机配置协议版本 4(DHCPv4)的快速提交选项
4174	建议标准	用于因特网存储名字服务的 IPv4 动态主机配置协议(DHCP)
4243	建议标准	动态主机配置协议(DHCP)中继代理选项的厂商特定信息子选项
4280	建议标准	用于广播和组播控制服务器的动态主机配置协议(DHCP)选项
4361	建议标准	用于动态主机配置协议版本 4(DHCPv4)节点特定的客户端标识符
4388	建议标准	动态主机配置协议(DHCP)Leasequery(租赁查询)
4390	建议标准	InfiniBand 之上的动态主机配置协议(DHCP)
4578	信息型的	用于 Intel 提前启动执行环境(PXE)的动态主机配置协议(DHCP)选项
4702	建议标准	动态主机配置协议(DHCP)客户端完全合格的域名(FQDN)选项
4703	建议标准	动态主机配置协议(DHCP)客户端间完全合格域名(FQDN)冲突解决
4776	建议标准	用于市政地址配置信息的动态主机配置协议(DHCPv4 和 DHCPv6)
4833	建议标准	DHCP 的时区选项
5010	建议标准	动态主机配置协议版本 4(DHCPv4)中继代理标志子选项

（续）

RFC	状态	标　　题
5071	信息型的	PXELINUX所用的动态主机配置协议选项
5107	建议标准	DHCP服务器标识符重写子选项
5192	建议标准	网络接入携带认证协议(PANA)认证代理的DHCP选项
5223	建议标准	使用动态主机配置协议(DHCP)发现位置到服务的转换(LoST)服务器
5417	建议标准	无线接入点控制和准备(CAPWAP)接入控制器DHCP选项
5460	建议标准	DHCPv6大块(Bulk)租赁查询
5678	标准跟踪	用于IEEE 802.21移动性服务(MoS)发现的动态主机配置协议(DHCPv4和DHCPv6)选项
5859	信息型的	DHCPv4的TFTP服务器地址选项

DHCPv6协议各RFC

RFC	状态	标　　题
3315	建议标准	用于IPv6的动态主机配置协议(DHCPv6)
3319	建议标准	会话初始协议(SIP)服务器的动态主机配置协议(DHCPv6)选项
3633	建议标准	动态主机配置协议(DHCP)版本6的IPv6前缀选项
3646	建议标准	用于IPv6的动态主机配置协议(DHCPv6)的DNS配置选项
3736	建议标准	用于IPv6的无状态动态主机配置协议(DHCP)服务
3769	信息型的	IPv6前缀委派的需求
3898	建议标准	用于IPv6的动态主机配置协议(DHCPv6)的网络信息服务(NIS)配置选项
4075	建议标准	DHCPv6的简单网络时间协议(SNTP)配置选项
4242	建议标准	用于IPv6的动态主机配置协议(DHCPv6)信息刷新时间选项
4580	建议标准	用于IPv6的动态主机配置协议(DHCPv6)中继代理订户-ID选项
4649	建议标准	用于IPv6的动态主机配置协议(DHCPv6)中继代理远程-ID选项
4703	建议标准	动态主机配置协议(DHCP)客户端间完全合格域名(FQDN)冲突解决
4704	建议标准	用于IPv6的动态主机配置协议(DHCPv6)客户端完全合格域名(FQDN)选项
4776	建议标准	用于市政地址配置信息的动态主机配置协议(DHCPv4和DHCPv6)选项
4833	建议标准	DHCP的时区选项
4994	建议标准	DHCPv6中继代理回声请求选项
5007	建议标准	DHCPv6租赁请求
5192	建议标准	网络接入携带认证协议(PANA)认证代理的DHCP选项
5223	建议标准	使用动态主机配置协议(DHCP)发现位置到服务的转换(LoST)服务器
5460	建议标准	DHCPv6大块地址租赁查询
5678	标准跟踪	用于IEEE 802.21移动性服务(MoS)发现的动态主机配置协议(DHCPv4和DHCPv6)
5908	标准跟踪	DHCPv6的网络时间协议(NTP)服务器选项

DNS 协议各 RFC

RFC	状态	标题
1034	标准	域名——概念和设施
1035	标准	域名——实现和规范
1101	未知	网络名和其他类型的 DNS 编码
1183	试验型的	新的 DNS RR 定义
1464	试验型的	使用域名系统来存储任意字符串属性
1480	信息型的	US(美国)域
1591	信息型的	域名系统结构和委派
1706	信息型的	DNS NSAP 资源记录
1712	信息型的	地理位置的 DNS 编码
1876	试验型的	在域名系统中表示位置信息的一种方法
1982	建议标准	序列号算术
1996	建议标准	区域变更提示通知的一种机制(DNS NOTIFY)
2136	建议标准	域名系统中的动态更新(DNS UPDATE)
2163	建议标准	使用因特网 DNS 来分发 MIXER 吻合性全球地址映射(MCGAM)
2181	建议标准	DNS 规范澄清说明
2182	BCP 16	辅助 DNS 服务器的选择和运行
2219	BCP 19	为网络服务使用 DNS 别名
2308	建议标准	DNS 查询的负面缓存(DNS NCACHE)
2317	BCP 20	无类 IN - ADDR. ARPA 委派
2536	建议标准	域名系统中(DNS)的 DSA KEY 和 SIG
2539	建议标准	域名系统(DNS)中存储 Diffie - Hellman 密钥
2540	试验型的	分离域名系统(DNS)信息
2671	建议标准	DNS 的扩展机制(EDNS0)
2672	建议标准	非终端 DNS 名重定向
2673	试验型的	域名系统中的二进制标签
2782	建议标准	指定位置服务的一条 DNS RR(DNS SRV)
2870	BCP 40	根名服务器运行需求
2874	试验型的	支持 IPv6 地址汇聚和重新编址的 DNS 扩展
3123	试验型的	地址前缀列表的一个 DNS RR 类型(APL RR)
3258	信息型的	通过共享的单播地址分发权威名字服务器
3363	信息型的	在域名系统(DNS)中表示因特网协议版本 6(IPv6)地址
3364	信息型的	因特网协议版本 6(IPv6)域名系统(DNS)支持中的折中考虑
3425	建议标准	过时的 IQUERY

（续）

RFC	状态	标 题
3467	信息型的	域名系统(DNS)的角色
3490	建议标准	应用中的国际化域名(IDNA)
3491	建议标准	Nameprep:国际化域名(IDN)的一个 Stringprep Profile
3492	建议标准	Punycode(次要代码):应用中国际化域名的 Unicode 的一个 Bootstring 编码
3596	草案标准	支持 IP 版本 6 的 DNS 扩展
3597	建议标准	未知 DNS 资源记录(RR)类型的处理
3681	BCP 80	E. F. F. 3. IP6. ARPA. 的委派
3901	BCP 91	DNS IPv6 传输运行指南
4074	信息型的	针对 IPv6 地址 DNS 查询的常见错误行为
4159	BCP 109	反对使用“ip6. int”
4183	信息型的	网络和网关 DNS 解析的一种建议方案
4185	信息型的	DNS 顶级域(TLD)名字的国家和本地字符(character)
4290	信息型的	国际化域名(IDN)注册的建议实践
4343	建议标准	域名系统(DNS)大小写不敏感概念澄清
4367	信息型的	名字中是什么:有关 DNS 名字的错误假定
4406	试验型的	发送者 ID:认证电子邮件
4406	试验型的	电子邮件地址中据称负责的地址
4407	试验型的	电子邮件中域的授权使用的发送者策略框架(SPF)版本 1
4472	信息型的	IPv6 DNS 的运行考虑和问题
4592	建议标准	域名系统中通配符的角色
4690	信息型的	国际化名字(IDN)的评述和建议
4697	BCP 123	观察到的 DNS 解析错误行为
4701	建议标准	用于编码主机配置协议(DHCP)信息的一种 DNS 资源记录(RR)(DHCID RR)
4892	信息型的	对识别一个名字服务器实例机制的需求
5001	建议标准	DNS 名字服务标识符(NSID)选项
5158	信息型的	6to4 DNS 委派规范
5205	试验型的	主机身份协议(HIP)域名系统(DNS)扩展[HIP RR]
5395	BCP 42	域名系统(DNS)IANA 考虑
5507	信息型的	当扩展 DNS 时的设计选择
5679	建议标准	使用 DNS 定位 IEEE 802.21 移动性服务
5855	BCP 155	IPv4 和 IPv6 反向区域的名字服务器
5864	标准跟踪	AFS 的 DNS SRV 资源记录

（续）

RFC	状态	标　　题
5890	标准跟踪	应用的国际化域名（IDNA）：定义和文档框架
5891	标准跟踪	应用的国际化域名（IDNA）：协议
5892	标准跟踪	应用的 Unicode 码点和国际化域名（IDNA）
5893	标准跟踪	用于应用的国际化域名（IDNA）的从右到左脚本
5894	标准跟踪	应用的国际化域名（IDNA）：背景、解释和合理性论据
5936	标准跟踪	DNS 区域传输协议（AXFR）

DNS 安全有关的各 RFC

RFC	状态	标　　题
2230	信息型的	DNS 的密钥交换委派记录
2845	建议标准	DNS 的安全密钥事务认证（TSIG）
2930	建议标准	DNS 的安全密钥建立（TKEY RR）
2931	建议标准	DNS 请求和事务性签名（SIG（0））
3007	建议标准	安全的域名系统（DNS）动态更新
3110	建议标准	域名系统（DNS）中 RSA/SHA－1 SIG 和 RSA KEY
3645	建议标准	用于 DNS 密码密钥事务认证的通用安全服务算法（GSS－TSIG）
3833	信息型的	域名系统（DNS）的威胁分析
4033	建议标准	DNS 安全导论和需求
4034	建议标准	DNS 安全扩展的资源记录
4035	建议标准	DNS 安全扩展的协议修改
4255	建议标准	使用 DNS 来安全地发布安全外壳（SSH）密钥指纹
4398	建议标准	在域名系统（DNS）中存储证书
4431	信息型的	DNSSEC 旁查（Lookaside）核验（DLV）资源记录
4470	建议标准	最小化覆盖 NSEC 记录和 DNSSEC 在线签名
4509	建议标准	DNSSEC 委派签名人（DS）资源记录（RR）中使用 SHA－256
4641	信息型的	DNSSEC 运行实践
4686	信息型的	促进域密钥识别邮件（DKIM）的威胁分析
4871	建议标准	域密钥识别邮件（DKIM）签名
4955	建议标准	DNS 安全（DNSSEC）试验
4956	试验型的	DNS 安全（DNSSEC）参与
4986	信息型的	与 DNS 安全（DNSSEC）信任锚点轮转有关的需求
5011	建议标准	DNS 安全（DNSSEC）信任锚点的自动化更新
5016	信息型的	域密钥识别邮件（DKIM）签名实践协议的需求
5074	信息型的	DNSSEC 旁查核验（DLV）

（续）

RFC	状态	标　　题
5155	建议标准	DNS 安全（DNSSEC）哈希经认证的存在性拒绝［NSEC3，NSEC3PARAM］
5358	BCP 140	防止在反射器攻击中递归使用名字服务器
5452	标准跟踪	使 DNS 对伪造答案更加具有抑制性的措施
5585	信息型的	域密钥识别邮件（DKIM）服务综述
5617	标准跟踪	域密钥识别邮件（DKIM）作者域签名实践（ADSP）
5672	标准跟踪	RFC 4871 域密钥识别邮件（DKIM）签名——更新
5702	建议标准	在 DNSSEC 的 DNSKEY 和 RRSIG 资源记录中与 RSA 一起使用 SHA－2 算法
5782	信息型的	DNS 黑名单和白名单
5863	信息型的	域密钥识别邮件（DKIM）开发、部署和运行
5933	标准跟踪	在 DNSSEC 的 DNSKEY 和 RRSIG 资源记录中使用 GOST 签名算法

DNS ENUM 有关的各 RFC

RFC	状态	标　　题
2916	建议标准	E.164 号和 DNS
3245	信息型的	电话号码映射（ENUM）的历史和背景…
3403	建议标准	动态委派发现系统（DDDS）第三部分：域名系统（DNS）数据库［NAPTR RR］
3761	建议标准	E.164 到统一资源标识符（URI）动态委派发现系统（DDDS）应用（ENUM）
3762	建议标准	针对 H.323 的电话号码映射（ENUM）服务注册
3764	建议标准	针对会话初始协议（SIP）地址记录的 ENUM 服务注册
3824	信息型的	会话初始协议（SIP）中使用 E.164 号码
3953	建议标准	针对在席服务的电话号码映射（ENUM）服务注册
3958	建议标准	使用 SRV RR 和动态委派发现系统（DDDS）的基于域的应用服务定位
4114	建议标准	可扩展信息准备提供协议（EPP）的 E.164 号码映射
4725	信息型的	ENUM 核验架构
4759	建议标准	“tel”URI 的 ENUM Dip 指示器参数
4848	建议标准	使用 URI 和动态委派发现协议（DDDS）的基于域的应用服务定位
5067	信息型的	基础设施 ENUM 需求
5483	信息型的	ENUM 实现问题和经验
5526	信息型的	基础设施 ENUM 的 E.164 到统一资源标识符（DDDS）应用
5527	信息型的	在 e164.arpa 树中组合用户和基础设施 ENUM

管理或运行方面的各 RFC

RFC	状态	标 题
1713	信息型的	DNS 调试的工具
1912	信息型的	常见 DNS 错误
2151	信息型的	因特网以及 TCP/IP 工具和设施基础
2606	BCP 32	保留的顶级 DNS 名字
3172	BCP 52	地址和路由参数区(Area)域(Domain)(“arpa”)的管理指南和运行需求
5157	信息型的	网络扫描的 IPv6 隐含意义

国际视野 科技前沿

国际信息工程先进技术译丛

《IP地址管理原理与实践》
《自组织网络：GSM，UMTS和LTS的自规划、自优化和自愈合》
《UMTS中的LTE：向LTE-Advanced演进》（原书第2版）
《UMTS中的WCDMA-HSPA演进及LTE》（原书第5版）
《LTE自组织网络(SON)：网络管理自动化提升运维效率》
《实现吉比特传输的60GHz无线通信技术》
《内容分发网络》
《无线Mesh网络架构与协议》
《UMTS蜂窝系统的QoS与QoE管理》
《半导体制造与过程控制基础》
《下一代移动系统：3G/B3G》
《IMS:IP多媒体概念和服务》（原书第2版）
《下一代无线系统与网络》
《深入浅出UMTS无线网络建模、规划与自动优化：理论与实践》
《HSDPA/HSUPA技术与系统设计——第三代移动通信系统宽带无线接入》
《无线传感器及元器件：网络、设计与应用》
《印制电路板——设计、制造、装配与测试》
《IPTV与网络视频：拓展广播电视的应用范围》
《多电压CMOS电路设计》
《微电子技术原理、设计与应用》
《蜂窝网络高级规划与优化2G/2.5G/3G/…向4G的演进》
《基于蜂窝系统的IMS——融合电信领域的VoIP演进》
《无线网络中的合作原理与应用》
《环境网络：支持下一代无线业务的多域协同网络》
《基于射频工程的UMTS空中接口设计与网络运行》
《未来UMTS的体系结构与业务平台：全IP的3G CDMA网络》
《UMTS-HSDPA系统的TCP性能》
《宽带无线通信中的空时编码》
《数字图像处理》（原书第4版）
《基于4G系统的移动服务技术》
《吉规模集成电路互连工艺及设计》
《高性能微处理器电路设计》

WILEY



上架指导 工业技术/信息技术

ISBN 978-7-111-40870-3 定价：89.80元

[General Information]
书名=IP地址管理原理与实践
页数=343
SS号=13221397